Ára: 30 Ft

μ mikro számítógép magazin

MESTERSÉGES ÉRTELEM

1987/6

DATASAVER—16 VIDEO STREAMER

a CONTROLL -tól

VIDEO + SZÁMÍTÓGÉP = GYORS, EGYSZERŰ, TAKARÉKOS ADATMENTÉS

CONTROLL

ELEKTRONIKAI ÉS SZÁMÍTÁSTECHNIKAI KISSZÖVETKEZET

BUDAPEST II., Szász Károly u. 2.
158-428, 158-430

SZAKÜZLET: XIII., Visegrádi u. 6. 128-064

5. ÉVFOLYAM
1987/6. SZÁM

μ mikro számítógép magazin

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG ÉS A KISZ KÖZPONTI BIZOTTSÁG LAPJA

A kiadvány a Tudományszervezési és Informatikai Intézettel együttműködve készül

A szerkesztőbizottság vezetője:
Kovács Győző

E számunkat szerkesztették:

Bakos Tamás
(programozástechnika)

Broczkó Péter
(hírek)

Kovács Győző
(levelezés)

Lindner László
(sakkprogramozás)

Petróczy Judit
(könyvek)

Simonyi Endre
(klub)

Varga András
(iskola-számítógép)

**Címképünk:
Ramocsai Imri munkája**

Felelős szerkesztő:
Könyves Tóth Pál

Szerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 154-250

Levélcím:
1371 Budapest
Pf. 433.

Kiadja az Ifjúsági Lap- és Könyvkiadó Vállalat

Felelős kiadó:
dr. Petrus György
igazgató

Kiadóhivatal:
1065 Budapest, Révay u. 16.
Telefon: 116-660

Terjeszti a Magyar Posta
Előfizethető a hírlapkézbesítő hivataloknál
és a Posta Hírlapelőfizetési és Lapellátási Irodáján
(1900 Budapest V., József nádor tér 1.)
vagy átutalással a 215—96 162 pénzforgalmi jelzőszámra.

Megjelenik havonta
Egy szám ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft
Külföldön terjeszti a Kultúra,
1389 Budapest, pf. 149.
és a Magyar Média
1932 Budapest, pf. 279.
86—0253

Szikra Lapnyomda
Budapest (87-0659)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

Tartalom

Két világ 2
Házi- és iskolaszámítógép-mérleg 9
ATARI kontra Commodore 64 11
Mesterséges értelem II. 15
Adok-veszek-cserélek 21
μ'87 25
Vigyázat! Tolvaj! 34
Olvastunk . . . 36
Einsteinnel vitázva 42
μINFORM 45

ISKOLA — SZÁMÍTÓGÉP

Nemes Tihamér Országos Középiskolai Tanulmányi Verseny 3
OKTA-TOTÓ 6

DIÁKROVAT

Képernyősűrítő 5
Szalagújság 6
Magyarul — hibáztál! 6
Autostart kazettára 7

PROGRAMOZÁSTECHNIKA

Programozzunk PASCAL-ban: scroll 19
BASIC és gépi kód 22
BASIC igényesnek, extrákkal eladó 23
Kritika és önkritika 24
RENUMBER 29
Teknősbéka-grafika II. 31
Folyamatábrák 32

μKLUB

Integrált szoftver 38
C64 programok átírása C128-ra 41
Ki ad magyarázatot? 41

SAKKPROGRAMOZÁS

Bitek és figurák 44

AZ OLVASÓ ÍRJA 46

KÖNYVEK 47

HÍREK, ÉRDEKESSÉGEK 48

„Nem számít, éltek-e valóban őseink is a földön, nagyobb csoda, hogy itt, valamikor teremtőik lehettünk."
(Fodor András: Visnu a barlangban)

Két világ

Tán nem haragszanak meg az olvasók, ha ismét a tapasztalataimról számolok be két konferenciával kapcsolatban. Közeli volt a tárgyuk — az informatika tanítása, illetve a számítógéppel segített oktatás, amelyet hívtak már CAI-nek (Computer Aided Instruction), azután CAL-nek (Computer Aided Learning), de manapság inkább CBT-nek (Computer Based Training) neveznek —, de minden egyéb nagyon is különböző. Az egyikre *11 gazdaságilag fejlett országból* jöttek el a szakemberek: a hollandiai székhelyű Courseware Group (CG) szervezte, és Zaandamban tartották. A másik kifejezetten munkakonferencia volt, kezdeményezője az Indian Institute of Technology (Bombay), és az évenkénti IFIP TC 3 (Oktatási Műszaki Bizottság) ülését követte. A bizottság 13 nemzeti képviselőjén kívül a környező 11 országból meghívott szakemberek elsősorban ezeknek az országoknak a nézőpontjából tárgyaltak.

Hollandiában a téma „The Courseware Industry" (szabad fordításban: az oktatásianyagkészítő ipar a világban). A négy fölkért előadó: A. Wujcik (USA), L. Steels (Belgium), M.M. Geerling (Hollandia) és G. Holland (Nagy-Britannia) a CBT helyzetéről, jövőjéről, a CBT-ben rejlő üzleti lehetőségről szólt. Kétségtelen, hogy az igény a számítógéppel segített oktatás iránt az USA-ban és a nyugat-európai országokban is növekvőben van, főképp mert igen sok ember — gyakorlatilag a munkavállalók — folyamatos informatikai képzésére (továbbképzésére és átképzésére) van szükség. Ezzel a hagyományos iskolarendszer nem tud megbirkózni, de tulajdonképpen nem is ez a feladata: egyik országban sincs elég tanterem, elég oktatói kapacitás, hogy a tömegképzést hagyományosan megoldják.

A CBT mellett még egy nagyon súlyos érv szól, nevezetesen: ez az „iskola" a „tanulóhoz" igazodik.

Szinte mindenki szerint a CBT technológiát ma elsősorban a felnőttoktatásban lehet és kell használni, ti. a vállalatok már felismerték, hogy informatikában képzett emberek nélkül nem lehet eredményt elérni — ezért áldoznak is a CBT-re, és nem is keveset. Az iskoláknak viszont többnyire nincs pénzük jó és technologizált tananyag készítésére; megveszik a számítógépeket (az USA-ban például 1,5 millió iskolai számítógép működik), de a legfontosabbra, a courseware-re már nem telik. Hollandiában mostanság gyökeres változásban reménykednek, mert az állami költségvetés a közeljövőben jelentős összegeket juttat számítógépes kurzusok kidolgozására.

Visszatérve a felnőttoktatásra, ma Európában a CBT-ben élenjáró országok: Nagy-Britannia, Franciaország és Hollandia; az oktatási programokat pedig részben az érdekelt vállalatok és intézmények, részben külső cégek — mint például a CG is — fejlesztik. Megrendelőik a bankok, a pénzintézetek és a biztosítótársaságok, de az utóbbi időben multinacionális cégeknél — az IBM, a Siemens, a Nixdorf, az Olivetti, a Fiat, az ENI stb. — is egyre többet fordítanak e korszerű „újratermelési módra".

Konkrét adatokat közölt A. Wujcik asszony, az USA-ból, a számítógéppel segített oktatásról. 1986-ban a 100-nál több alkalmazottas vállalatok 53,1 százaléka használt e célra számítógépet, 32,1 százaléka pedig CBT-t. A kisebb cégek kevesebbre jutottak ez idáig: a 100-tól 500 dolgozót foglalkoztatóknak csak a 30 százaléka, míg a 10 ezer főnél nagyobbaknak 66,7 százaléka részesíti előnyben a képzésnél a számítógépet.

Hogy az oktatási forma és a számítógépes kultúra mennyire összefügg, az is mutatja, hogy míg a 100-tól 500 fős szervezetek 41,5 százaléka vesz igénybe rendszeresen adatbázisokat és 42,3 százalékuk használ szövegfeldolgozó és grafikus rendszereket, a 10 ezernél nagyobbaknál ugyanez az arány mindkét esetben 83,3 százalék. De még az USA-ban is az 500 alatti kategória 13,3 százalékának és még a 10 ezernél több alkalmazottat foglalkoztató vállalatok 6,4 százalékának sem kellett 1986-ban számítógép!

Az ismeretközlés tartalmi részét nézve az 50 főснél nagyobb cégeknél az első helyen áll a vezetési ismeretek oktatása (76,5 százalékuknál). Alig kevesebben tanítanak különféle műszaki ismereteket számítógéppel (69,8 százalék), és ennél csekélyebb figyelmet fordítanak az adminisztráció (58,2 százalék) vagy a kereskedelmi tudnivalók (43,3 százalék) megújítására. 52,6 százalékuknál a tananyag az informatikai alapokra szorítkozik.

Az amerikai cégek 45 százalékánál a számítógépes oktatás eszköze a nagyszámítógép és a hálózat, 33 százalékuknál a minigép és 82 százalékuknál a PC. Tavaly az Apple Macintosh gépre kidolgoztak egy sor igen hatékony oktatóprogramot; ebből az oktatástechnológiával foglalkozó szakemberek arra következtetnek, hogy a jövőben az IBM PC-k (és klónok) viszonylatában az arány eltolódik a Macintosh javára.

Az előadásokból kitetszett, hogy courseware készítése világszerte a Plato hagyományait követő szerzői rendszer (authoring system) alapján folyik. A CG által bemutatott hatékony amerikai rendszer továbbfejlesztett változatával — a Ten CORE—Pet-tel — a számítástechnikában nem járatos tanárok is *jó* tananyagot tudnak készíteni. Még ma általában egy valamirevaló oktatóprogram megvalósításához legalább két szakember kell: egyik a tananyagot ismeri és állítja össze, a másik a számítógépes programot csinálja (ez sokszor nem felel meg a pedagógiai követelményeknek). A CBT törekvése arra irányul, hogy a tanárok olyan szoftvereszközöket kapjanak, amelyekkel segítség nélkül is vállalkozhatnak oktatásirendszer-készítésre.

Sok szó esett az IAV (Interactive Video) rendszernek az oktatási alkalmazásáról. Professzionális személyi számítógépekhez igen nagy kapacitású (800 Mbájt) videolemezt kapcsolnak: ezen szöveget, álló- és mozgóképeket lehet tárolni. A videolemezről származó kép együtt jelenhet meg a számítógép memóriájában tárolt szöveges vagy rajzos információval, így módon hasonlíthatatlanul hatékony a rendszer. Bár videolemezt készíttetni csak 1000 példány fölött gazdaságos, egyre többen választják ezt a megoldást: az USA-ban 1985-ben az 50 főснél nagyobb munkáltatók 11,6 százaléka, 1986-ban már 15,5 százaléka rendelkezett IAV rendszerekkel: 1987-re 20,2 százalékot becsülnek.

Foglalkozzunk most a mesterséges intelligencia oktatási szerepével. A jelenlegi kutatások főleg intelligens konzultációs rendszerek (Intelligent Tutorial Systems) létrehozására irányulnak.

A dinamikusan fejlődő CBT piacon Nyugat-Európa bizonyos lemaradással követi a tengerentúliakat. Európában a „konzervatívok" egyik legtöbbet hangoztatott ellenérve, hogy a CBT elemberteleníti, mechanizálja az oktatás folyamatát, és ezért távlatban a hatékonysága megkérdőjelezhető.

A bombayi rendezvény egészen más környezetben, a „National Issues on Computers in Education", azaz szabad fordításban: „A számítógépek oktatásban való alkalmazásának nemzeti eredményei" címmel marad emlékezetes számomra.

Az ott lévők nagyobbik része Indiából és a közeli országokból (Pakisztán, Bangladesh, Korea, Srí-Lanka, Malaysia, Hongkong, Szingapúr) jött, csak nagyon kevesen voltunk Európából, az USA-ból és Ausztráliából. A problémákat az egyes témák nem odavalósi előadói részben a saját környezetükben tapasztaltak szerint, részben pedig a fejlődő országokra adaptálva próbálták meg kifejteni, ez utóbbit — a tényleges helyzet elégtelen ismerete miatt — kevés sikerrel.

Bár az ázsiai országok zöme igen súlyos szociális gondokkal küzd (éhínség, munkanélküliség, nemzetiségi ellentétek stb.), tanulságos, hogy ennek ellenére nagyon komolyan foglalkoznak az informatika társadalmi méretű elterjesztésével. Teszik ezt akkor is, ha még egyáltalán nincs meg a feltétele, hogy az informatizált társadalmat belátható időn belül legalább megközelítsék.

A 700 milliós Indiában ma kb. 8 ezren tanulnak informatikát, de hiába is tanulnának többen, se India — ahol van számítógépgyártás —, de ma még a fejlődő országok többsége sem képes megfelelő mennyiségű számítógépet vásárolni. Részben mert a számítógép viszonylag drága, részben mert más, a lakosságnak fontosabbak — például élelmiszer-termelési technológiák — beszerzése előbbre való. Néhányan mégis másképp érveltek: a fejlődő országok nem mulaszthatják el az informatika széles körű terjesztését: ma még közeli az esély a versenyben, mert — ahogy mondták — a nyolcvanas években a vezető ipari országok kivételével „az informatikában csak fejlődő országok vannak".

Az informatika iskolai színre lépése szinte valamennyi ázsiai országban észlelhető. Az európai gyakorlattal szemben először arra törekszenek, hogy néhány iskolában a tanárok ismerjék meg a számítógépeket, és ezután kezdjék el — hagyományosan: példák bemutatásával — az alkalmazás oktatását. Képviselőik keveset mondtak el saját tapasztalataikból — úgy látszott, hogy nincsenek ilyenek —, annál jobban érdekelték őket a másutt már bevált módszerek.

Megdöbbentő, hogy — a hiányos orvosi ellátás miatt — ezekben az országokban mennyivel több a valamilyen betegség miatt hátrányos helyzetű ember, mint például Európában. A térség „gazdagjai" (Hongkong, Szingapúr) a fizikailag hátrányos helyzetűek részére különleges — úgymond személyre szóló — programokat dolgoztak ki, amelyekben komoly szerepet kap az informatika. A konferencia egyik indiai résztvevője biztató kísérletről számolt be: kommunikációra tanítanak süketnémákat számítógéppel.

Akadtak persze elmaraszta lő vélemények is, hiszen szóhoz jutottak a munkakonferencián a mai világméretű informatikaképzési hullámot nem nagyon támogató szakemberek is. Egyikük élesen vetette föl, hogy ha már Indiában is enynyire szükség van informatikai szakemberekre, akkor miért nem a 300 ezer munkanélküli értelmiségit képezik át először ahelyett, hogy óriási pénzekkel a hátrányos helyzetűeket segítik. Egy másik „ellenzéki" szerint a térség több „szegény" országában, főleg falun az a gond, hogy még elfogadható iskolaépület sincs, és nem pedig, hogy használnak-e számítógépet.

Utoljára hagytam a távtanulásról és az Open Universityről (nyílt egyetem) folytatott eszmecserét. A témáról két indiai szakemberrel referáltunk a résztvevők igen jóleső érdeklődése mellett. A hozzászólók megerősítették, hogy a távtanulási technológia és a nyílt egyetem talán az egyetlen lehetőség arra, hogy a sok más diszciplinát magán hordozó informatika széleskörűen elterjedjen. Indiában egy éve indult a nyílt egyetem; meglehetősen zürösen. A szokásos akadályok voltak a legsúlyosabbak: a hagyományos egyetemek féltékenysége, illetve a kooperatív készség hiánya, az előítéletek. Elterjedt ti., hogy a nyílt egyetem nem ad olyan képesítést, mint a rendes, ez a diploma értéktelenebb lesz, mint az; nem érdemes ide beiratkozni. Az Indiai Nemzeti Nyitott Egyetem nem rendelkezik ugyan olyan felszereléssel, mint néhány más egyetem, de elhatározásuk, hogy erős vizsgákkal szerzik meg a társadalmi elismerést. Egyetlen ténnyel senki sem vitatkozik: nincsen másik járható út a falusi emberek fölzárkóztatásában (India lakosságának 75 százaléka falusi környezetben él), se az ugyanígy problémás női továbbtanulásban (ez a társadalmi közeg az asszonyok helyét csakis a családban jelöli ki). Végül nem maradt nézeteltérés: a távtanulási technológia, illetve a nyílt egyetem a társadalmi méretű oktatás egyetlen lehetséges alternatívája.

Bombayben tehát — ellentétben a hollandiai konferenciával — kevés szó esett a technikai kérdésekről, sokkal több a szociális és társadalmi vetületekről. Mi valahol — a két világ között — a félúton vagyunk, így van mit megszívlelnünk.

KOVÁCS GYŐZŐ

Nemes Tihamér Országos Középiskolai Tanulmányi Verseny

Ez a verseny a III. és IV. osztályos középiskolásoknak ad lehetőséget számítástechnikai ismereteik és képességeik összehasonlítására. A kétfordulós szervezés előnye, hogy sokan indulhatnak: az első fordulóban, melyet az iskolákban tartanak meg, a fiatalok analizáló képességükről adnak számot. Az itt eredményesen szereplő, vagyis a verseny közben kialakult szintet elérő „jobbak" jutnak tovább; a második menetben már „csak" egyetlen, nagyobb léleqzetű feladatot kell megoldaniuk. Ez az erőpróba szintetizáló, konstruktív tulajdonságaikra épít, és a megoldáshoz számítógépet használhatnak.

Az első fordulóban mintegy 250 iskola 1900 tanulója vett részt. Az idén a 40 százalékos szint volt a kiesés elkerülésének minimumfeltétele; a versenybizottsághoz kb. 280 ilyen értékű dolgozat érkezett be.

Tapasztalatok szerint az idei feladatok gondolkoztatóbbak voltak az előző évinél, ám a megoldások nehézségi fokát is jól eltalálták a feladatok összeállítói. Emiatt a versenyzők mezőnye a dolgozatokban elért pontszámok alapján széthúzódott. Végül is a második fordulóba kerüléshez kb. 60 pont kellett, ami 85 százalékos szintnek felelt meg. Tavaly a továbbjutáshoz már 60 százalék is elég volt.

A versenybizottság végleges véleményére még várnunk kell, de az előzetes értékelés máris hasznosítható tapasztalatokkal szolgál. Az 1. feladat pontosabb megfogalmazásával a versenyzőket több lehetséges megoldásra lehetett volna ösztönözni. A 2. feladatnál a tanulók gyakran nem vették figyelembe, hogy az előző eredményt az automata a veremből ki is veszi. A feladat megfogalmazói erre szándékosan nem tértek ki, mert a veremfunkció ismeretére voltak kíváncsiak. A 4. feladat esetében azonban kifejezetten fel kellett volna hívni a tanulók figyelmét, hogy általánosításra törekedjenek. Ennek ellenére sok szép megoldás érkezett. A 8. feladat leírásába sajnos hiba csúszott: az A és B rész második sorában — a kisebb vagy egyenlő jel helyett — helyesen nagyobb vagy egyenlő jelnek kellett volna szerepelnie. Az elírásra a versenyzők általában rájöttek, így ez nem zavarta meg a verseny lebonyolítását.

A versenybizottság külön örömére a nehéznek tartott 10. feladatot a diákok jól értelmezték, és sok helyes megoldás született.

Az alábbi ÉRTÉKELÉSI ÚTMUTATÓ összeállításánál nem törekedhettek a teljeskörűségre, mert egyes feladatok megoldására számtalan lehetőség van. Az elbíráláshoz azonban minden helyes megoldást figyelembe vettek.

Most az első forduló feladatait és megoldásait ismertetjük.

1. forduló 1987. február 10.

1. Melyek a bemenő adatok, s milyennek kell lenniük, hogy a következő programrészlet jól működjön? A program egy számsorozatot úgy ír ki, hogy először a benne szereplő negatív elemeket írja, majd pedig a nemnegatívakat! Mi a szerepe az F1, F2 változóknak és a B(N) vektornak?

```
110 F1=0 : F2=0 : X=0
120 FOR I=1 TO N
130 IF A(I)>=0 THEN B(I)=F2: F2=I
      ELSE B(I)=F1 : F1=I : IF B(I)=0 THEN X=I
140 NEXT I
150 B(X)=F2 : X=F1
160 IF X<>0 THEN PRINT A(X) : X=B(X) : GOTO 160
```

Elérhető pontszám: 12 pont

Megoldás:

N>=1 és egész. 2 pont
A(N) vektor, tartalmazzon legalább egy negatív elemet! 4 pont
F1 az A vektorbeli negatív elemek közül az utolsó, F2 a nemnegatívak közül az utolsó sorszáma lesz. 2 pont
(Ha az utolsót nem mondja, akkor 1 pont levonás.)
A B(N) vektor az A-beli elemek listába fűzésére szolgál a kívánt sorrendben. 2 pont
A negatívak és a nemnegatívak sorrendje is megfordul önmagukon belül- az eredetihez képest. 2 pont

2. Egy veremautomata a következőképpen működik: ha számot kap a bemenetén, azt egy verembe teszi, ha műveleti jelet, akkor azt a verem tetején levő számokkal elvégzi, majd az eredményt a verembe teszi (pl. A-B a következőképpen néz ki az automata nyelvén: A B -). A veremautomata bemenetén a következő sorozatot kapja (a számokat egymástól és a jelektől szóköz választja el):

1 3 + 2 5 4 * 8 - 6 / + -

Add meg, mi lesz a verem állapota az egyes jelek (adat vagy művelet) érkezése után. (Illetve a feldolgozás végén!

Elérhető pontszám: 5 pont

Megoldás:

1
1,3
4
4,2
4,2,5
4,2,5,4
4,2,20
4,2,20,8
4,2,12
4,2,12,6
4,2,2
4,4
8

Tévedésenként 1 pont levonás jár. 4-nél több hiba esetén 0 pont.

3. Mit csinálnak a következő rekurzív programok (X az angol ABC betűinek sorozata)?

```
F(X):
  Ha X üres akkor eredmény:=0
        különben Ha ( első(X) (A,E,I,O,U)-valamelyike )
                    akkor eredmény:=1+F(elsőnkívüli(X))
                    különben eredmény:=F(elsőnkívüli(X))
Függvény vége.

F(X):
  Ha X üres akkor eredmény:=IGAZ
        különben eredmény:= ( első(X) (A,E,I,O,U)-valamelyike )
                    és F(elsőnkívüli(X))
Függvény vége.

F(X):
  Ha elsőnkívüli(X) üres akkor eredmény:=első(X)
        különben Ha első(X)>F(elsőnkívüli(X))
                    akkor eredmény:=első(X)
                    különben eredmény:=F(elsőnkívüli(X))
Függvény vége.
```

Megjegyzés: Az első(X) függvény megadja az X karaktersorozat első tagját. Az elsőnkívüli(X) függvény megadja az X karaktersorozat tagjait a 2.-tól a végéig. Az és a szokásos logikai 'és' művelet.

Elérhető pontszám: 12 pont

Megoldás:

X magánhangzóinak száma 4 pont
X minden betűje magánhangzó-e 4 pont
X kódsorrendben (angol ABC-sorrend) legnagyobb eleme 4 pont

4. Mik a hibák a következő programban? Adj meg egy olyan bemenő adatot, amelyre helyesen működik!

```
10 INPUT "KEREK EGY EGESZ SZAMOT (>2)";N
15 IF N<3 OR N<>INT(N) THEN 10
20 K=0 : L=0
30 FOR I=2 TO SQR(N)
40   IF N/I=INT(N/I) THEN K=K+1
50   IF K=1 THEN L=I
60 NEXT I
70 PRINT "VALODI OSZTOK SZAMA=";K*2
80 PRINT "LEGKISEBB VALODI OSZTO=";L
90 STOP
```

Elérhető pontszám: 9 pont

Megoldás:

A: négyzetgyök(N)-et kétszer számolja az osztók számába. 2 pont
az 50-es sorban a feltétel az első és a második osztó megtalálása közötti összes számra teljesül (csak, ha az 1 osztó és a K=1, akkor lenne szabad az L=I utasítást végrehajtani). 2 pont
a 80-as sorban,ha N prímszám volt, akkor nincs legkisebb valódi osztó, azaz a kiírást csak L>0 esetben szabad elvégezni. 3 pont

B: minden olyan számra helyesen működik a program, amelynek a 2 és a 3 osztója, a szám négyzetgyöke nem osztó (nem egész), valamint maga a szám nem prímszám. 3 pont

5. A következő LOGO nyelvű program, adott n és f értékre, egy ábrát rajzol a képernyőre (az egyes utasításokat szóköz választja el egymástól).

FORWARD 2*n **RIGHT** f **FORWARD** n **LEFT** f/2 **BACK** n **LEFT** f/2 **BACK** n

Az utasítások jelentése (a LOGO-ban egy teknőcnek nevezett kurzor segítségével rajzolhatunk, amely az aktuális irányba előre vagy hátra képes lépni, valamint jobbra és balra tud fordulni):

FORWARD hossz - előre lép és vonalat húz
BACK hossz - hátra lép és vonalat húz
RIGHT fok - jobbra fordul helyben
LEFT fok - balra fordul helyben

Mit kell módosítani, ha azt szeretnénk elérni, hogy a kapott ábra az eredeti tükörképe legyen
A: az eredeti irányban a kezdőponton át húzott egyenesre,
B: az eredeti irányra merőleges, a kezdőponton át húzott egyenesre,
C: a kezdőpontra?
A végrehajtandó utasítások száma nem változhat! (Próbálj többféle megoldást adni!)

Elérhető pontszám: **12** pont

Megoldás:

VAGY

A: f helyére -f — A: LEFT helyére RIGHT és RIGHT helyére LEFT 3 pont
B: n helyére -n, valamint f helyére -f — B: LEFT helyére RIGHT és RIGHT helyére LEFT, valamint FORWARD helyére BACK és BACK helyére FORWARD 3 pont
C: n helyére -n — C: FORWARD helyére BACK és BACK helyére FORWARD 3 pont

MINDEN HELYEN

Ha az egyes feladatokra mindkét változatot megadja, akkor további 1-1 pont adható.

6. A következő programrészletek A és B egész számok (B<>0) hányadosának egészrészét határozzák meg és helyezik el az X változóban. Egyik sem működik tetszőleges egész számokra. Milyen feltételek mellett helyesek az egyes megoldások (a B<>0 feltételen kívül)?

```
A:  Osztás:
      Ciklus amíg B>1
        Ha A páros és B páros akkor A:=A/2 : B:=B/2
      Ciklus vége
      X:=A
    Eljárás vége.

B:  Osztás:
      X:=0
      Ciklus amíg abs(A)>=abs(B)
        A:=A-B : X:=X+1
      Ciklus vége
    Eljárás vége.
```

Elérhető pontszám: **5** pont

Megoldás:

A: B 2-hatvány és osztója A-nak 2 pont
B: A és B előjele megegyezik 3 pont

7. A következő gépi kódú program az A és a B regiszterben vár egy-egy egész számot. Mely esetben fejezi be a végrehajtást az IGEN címkénél illetve mely esetben a NEM címkénél (A,B>0)?

```
CIKLUS:
        CP      B               ; hasonlítsuk össze A-t a B-vel!
        JP      Z,IGEN          ; ugrás, ha A=B volt
        JP      M,NEM           ; ugrás, ha A<B volt
        BIT     0,A             ; A legkisebb helyiértékű bitje 1-es-e?
        JP      NZ,PTLANA       ; ugrás, ha 1-es volt
        BIT     0,B             ; B legkisebb helyiértékű bitje 1-es-e?
        JP      NZ,PTLANB       ; ugrás, ha 1-es volt
        SRL     A               ; A bitjeinek eggyel jobbra léptetése
                                ; a legmagasabb helyiértékre 0 lép be
        SRL     B               ; B bitjeinek eggyel jobbra léptetése
                                ; a legmagasabb helyiértékre 0 lép be
        JP      CIKLUS          ; ugrás a CIKLUS címkére
PTLANA:
        BIT     0,B             ; B legkisebb helyiértékű bitje 1-es-e?
        JP      Z,NEM           ; ugrás, ha 0-ás volt
        SUB     B               ; A:=A-B
        JP      CIKLUS          ; ugrás a CIKLUS címkére
PTLANB:
        SRL     A               ; A bitjeinek eggyel jobbra léptetése
                                ; a legmagasabb helyiértékre 0 lép be
        JP      CIKLUS          ; ugrás a CIKLUS címkére
IGEN:   ...
        ...
NEM:    ...
```

Elérhető pontszám: **10** pont

Megoldás:

Az IGEN címkén fejezi be a végrehajtást, ha B osztója A-nak, egyébként pedig a NEM címkén.
Részmegoldásként 1-1 pont adható a következő válaszokért:
- A=B esetén az IGEN címkén fejezi be,
- A<>B esetén az IGEN címkén fejezi be,
- B=1 esetén az IGEN címkén fejezi be,
- A<B esetén a NEM címkén fejezi be.

8. Egy tetszőleges F(X) folytonos függvény zérushelyét keressük E pontossággal az [A,B] intervallumon (abs(F(X))<=E tulajdonságú X-et keresünk). Milyen elégséges feltételeknek tegyen eleget az F(X) függvény az A és a B pontban, hogy az [A,B] intervallumban biztosan legyen legalább egy zérushelye és a programrészlet ezek közül egyet megtaláljon?

```
A:  K:=(A+B)/2
    Ciklus amíg abs(F(K))<=E
        Ha F(K)<0 akkor A:=K különben B:=K
        K:=(A+B)/2
    Ciklus vége
    Ki: K,F(K)

B:  K:=A-F(A)*(B-A)/(F(B)-F(A))
    Ciklus amíg abs(F(K))<=E
        Ha F(K)>F(A)<0 akkor B:=K különben A:=K
        K:=A-F(A)*(B-A)/(F(B)-F(A))
    Ciklus vége
    Ki: K,F(K)
```

Elérhető pontszám: **8** pont

Megoldás:

A: F(A)<=0 és F(B)>=0 3 pont
B: F(A)>F(B)<=0 4 pont
F(A)=F(B)=0 egyszerre nem lehet 1 pont

9. Írd át gyorsabb működésűre a következő programrészletet! (A program BASIC nyelvű legyen és egy sorban csak egy utasítás lehet.) A program H0 magasságból leeső M tömegű test helyzeti és mozgási energiáját számolja méterenként.

```
100 INPUT M,H0
110 DIM X(H0),Y(H0)
120 G=9.81
130 FOR H=H0 TO 0 STEP -1
140   EH=M*G*H
150   EM=M*G*H0-M*G*H
160   X(H)=EH
170   Y(H)=EM
180 NEXT H
```

Elérhető pontszám: **7** pont

Néhány lehetséges megoldás:

```
100 INPUT M,H0                  100 INPUT M,H0
110 DIM X(H0),Y(H0)             110 DIM X(H0),Y(H0)
120 G=9.81                      120 G=9.81
121 MG=M*G                      121 MG=M*G
122 E0=MG*H0                    122 EH=MG*H0
                                123 EM=0
130 FOR H=H0 TO 0 STEP -1       130 FOR H=H0 TO 0 STEP -1
140   X(H)=MG*H                 140   X(H)=EH
150   Y(H)=E0-X(H)              150   Y(H)=EM
                                160   EH=EH-MG
                                170   EM=EM+MG
180 NEXT H                      180 NEXT H
```

Pontszám: 6 minusz a ciklus belsejében levő szorzások száma, illetve +1 pont, ha a ciklusmaghan az értékadások, összeadások, kivonások, tömbindexelések együttes száma <= 8.

10. Pisti mikroszámítógépéhez kapott egy "egeret". Ez egy cigarettásdoboz méretű eszköz, amely az asztalon szabadon tologatható és elmozdulását egy vezetéken keresztül közli a számítógéppel. Az egér állapota az INP(63) utasítással kérdezhető le. Pisti írt egy rövid programot:

```
10 PRINT INP(63);
20 GOTO 10
```

A program elindítása után Pisti várt egy kicsit, majd lassan elmozdította az egeret először jobbra, majd balra, azután előre, végül hátra. Az irányváltások előtt picit mindig várt. A képernyőn a következő számsorozat jelent meg:

```
!<-- az egér áll -->!<---------- jobbra mozog ---------->!<---- áll ---->!
248 248 240 248 248 248 242 243 241 240 242 243 241 248 242 242 242 242 242

!<------ balra mozog ------>!<-- áll -->!<------ előre mozog ---------->!
248 241 243 242 248 241 243 243 243 243 251 255 247 243 251 255 247 243

!<---- áll ---->!<------ hátra mozog ---------->!<---- áll ---->!
243 243 243 243 247 255 251 243 247 255 251 243 247 247 247 247 247
```

Pisti ezután a következő programot írta (egyes részeket neked kell megírnod!):

```
10 X=10 : Y=10 : X0=0 : Y0=0
20 E=INP(63) : E1=E-16*INT(E/16) : Y1=INT(E1/4) : X1=E1-4*Y1
30 X9=X0+10+X1
40 IF feltétel1 THEN X=X+1 : GOTO 60 : REM jobbra
50 IF feltétel2 THEN X=X-1 : REM balra
60 Y9=Y0+10+Y1
70 IF feltétel3 THEN Y=Y+1 : GOTO 90 : REM előre
80 IF feltétel4 THEN Y=Y-1 : REM hátra
90 PLOT X,Y : REM kirajzolja az (X,Y) koordinátájú pontot
100 X0=X1 : Y0=Y1 : GOTO 20
```

A: Mit jelentenek az egyes bitek?
B: Mi a szerepe az (X,Y), (X0,Y0), (X1,Y1), (X9,Y9) változópároknak?
C: Mit csinál a program és mi jelenik meg a képernyőn?
D: Add meg a feltétel1, feltétel2, feltétel3, feltétel4 helyére beírandó, X9-től vagy Y9-től függő logikai kifejezéseket!

Elérhető pontszám: **15** pont

```
Megoldás:

   A: A beolvasott számok bináris alakban:
      248 = 1111 00 00      !          ^          Az utolsó két bit
      242 = 1111 00 10      !          !          változik, a többi
      243 = 1111 00 11      ! jobbra   ! balra    változatlan.
      241 = 1111 00 01      V          !

      243 = 1111 00 11      !          ^          A következő két bit
      251 = 1111 10 11      !          !          változik, a többi
      255 = 1111 11 11      ! előre    ! hátra    változatlan.
      247 = 1111 01 11      V          !

      A beolvasott byte felső 4 bitje állandóan 1.              1 pont
      Az alsó két bit az X irányú, a következő két bit
      az Y irányú elmozdulástól függő kódszót tartalmaz.        1 pont
      Az elmozdulás irányát a kódok sorrendje adja meg.         1 pont

   B: (X,Y)= az egér pozíciójának koordinátái).                 1 pont
      (X1,Y1)= az egér pillanatnyi állapotának 2-2 bites
               kódja.
      (X0,Y0)= az egér előző állapotának 2-2 bites kódja.       1 pont
      (X9,Y9)= az előző és a pillanatnyi állapotból
               számított decimális érték.                       1 pont

   C: A program követi az egér mozgását és pontsorozat-
      ként felrajzolja a bejárt utat a képernyőre.              1 pont
      A képernyőn a (10,10) pontból indul és nem ellen-
      őrzi, hogy a kirajzolandó pont a képernyőre esik-e.       1 pont

   D: feltétel1: X9=2 OR X9=23 OR X9=31 OR X9=10  (jobbra)
      feltétel2: X9=20 OR X9=32 OR X9=13 OR X9=1  (balra)
      feltétel3: Y9=2 OR Y9=23 OR Y9=31 OR Y9=10  (előre)
      feltétel4: Y9=20 OR Y9=32 OR Y9=13 OR Y9=1  (hátra)

      Ha egy feltételt hibátlanul ad meg:                       3 pont
      Ha az ellenkező irányú feltételt is hibátlanul
      adja meg:                                                 +2 pont
      A további két feltétel megadásáért feltételenként:        +1 pont


Az 1# feladat összpontszáma: 9%
```


SOFTWARE '88 VÁSÁR

Az Ipari Minisztérium és a Központi Statisztikai Hivatal védnöksége és támogatása mellett a SOFTWARE '88 kiállítás keretén belül megrendezzük a

SOFTWARE '88 VÁSÁRT

A vásárra olyan szoftvertermékek nevezését várjuk, melyek:

- **— működőképesek,**
- **— dokumentáltak,**
- **— placképesek.**

A szoftverek benevezésének előfeltétele, hogy azokat egy — a gyártótól és forgalmazótól független — cég vagy javasolt szakértő minősítse. A minősítés szempontjai és a szakértői jegyzék az előzetes jelentkezés alkalmával a rendezőségtől szerezhető be.

Az előzetes jelentkezéseket május, június hónapban várjuk:

SW '88 rendezőség, COMPORGAN RENDSZERHÁZ K. V., 1277 Budapest 29. Pf. 27. Telefon: 150-856. Ügyintéző: Szádeczky-Kardoss Ákos, Sári Gyuláné.

Nevezési díj: vállalatoknak és szövetkezeteknek 4000 Ft/termék, jogi személyiséggel nem rendelkezőknek 2000 Ft/termék.

A minősített termékek leadási határideje: 1987. szeptember 7. A vásári termékeket szakemberekből álló bizottság értékei, és a legkiválóbb terméket díjazza.

I. díj 100 000 Ft
II. díj 75 000 Ft
III. díj 50 000 Ft

1987. NOV. 10-13.

SOFTWARE'88
BUDAPEST
Hotel Duna Inter•Continental

Képernyő-sűrítő

DIÁK spectrum ROVAT

E program segítségével csökkenthetjük képernyőink helyfoglalását a tárban, ami különösen több képet tartalmazó játékprogramok esetében előnyös. A program két részből áll: az első a kép sűrítését, a második a visszatöltést végzi. A két rutint bárhova tölthetjük a tárban, mert csak relatív ugrásokat tartalmaznak. A sűrített kép a 40000-es címtől helyezkedik el. Ezt az értéket megváltoztathatjuk: az első rutinban a IX, a másodikban a HL regiszterértéket kell átírni. A listán az átírandó értékeket bekereteztük. Elöl áll a cím alacsony, mögötte a magas helyiértékű bájtja.

A sűrítő rutint célszerű LET A=USR kezdőcímmel indítani, így az A változóban megkapjuk a lerövidített kép hosszát. Kazettára mentésnél a visszatöltő programot a rövidített kép mögé tegyük be, mert akkor a képpel együtt egy részben vihetjük szalagra.

Sűrítésre a program 430-as, visszatöltésre 440-es sora mutat példát.

FETSER IGNÁC

```
109 REM  **********************
110 REM  *   SURITO RUTIN     *
111 REM  **********************
120 FOR  F=32000 TO 32088
130 READ A:POKE F,A
140 NEXT F
150 DATA  33,  0, 64,  1,  0, 27
160 DATA 221, 33, 64,156,126, 95
170 DATA  11,120,177, 40, 59,123
180 DATA  35,190, 40,  7,221,119
190 DATA   0,221, 35, 24,237,221
200 DATA 119,  0,221, 35,221,119
210 DATA   0,221, 35, 22,  2,126
220 DATA  95, 35, 11,120,177, 40
230 DATA  27,123,190, 32,  8, 20
240 DATA 122,254,255, 40,  9, 24
250 DATA 236,221,114,  0,221, 35
260 DATA  24,198, 35, 11,120,177
270 DATA  40,  2, 24,241, 55, 63
280 DATA   1, 64,156,221,229,225
290 DATA 237, 66,229,193,201
299 REM  **********************
300 REM  * VISSZATOLTO RUTIN *
301 REM  **********************
310 DATA  33,  0, 64,221, 33, 64
320 DATA 156,  1,  0, 27,221,126
330 DATA   0, 95,221, 35,221,190
340 DATA   0, 40,  8,119, 35, 11
350 DATA 120,177, 32,238,201,221
360 DATA  35,221, 86,  0,115, 35
370 DATA 111,120,177,200, 21, 32
380 DATA 247,221, 35, 24,219
400 FOR  F=32100 TO 32146
410 READ A:POKE F,A
420 NEXT F
430 REM  LET A=USR 32000:SURIT
440 REM RANDOMIZE USR 32100:VISSZATOLT
```

DIÁK spectrum ROVAT

Szalagújság

Ez a kis rutin jó szolgálatot tesz nagyobb programok betéteként, ha tudatni akarunk valamit a felhasználóval. Sebessége miatt játékprogramokban is igen hasznos lehet. A könnyebbség kedvéért a betöltő eljárással együtt közöljük a kódot. *(1. lista)*

A program az A$-ban megadott szöveget szalagújságszerűen görgeti. A program a gép maszkolható megszakítását kihasználva működik. Ez jelentősen megkönnyíti a vele való munkát, mivel elég egyszerű elindítani, és utána már nincs is teendő. Feltűnő, hogy a rutin függőleges irányban kétszeresére nagyított karaktereket készít, és ehhez nem kell külön karakterkészletet begépelni, mert a 15360-as címtől elhelyezkedő adatokat kezeli. A megszakítás RANDOMIZE USR 320884-gyel indítható, leállítani pedig RANDOMIZE USR 32091-gyel lehet. Érdekes hatásokat érhetünk el, ha beírjuk a következő parancspárokat:

POKE 32058,54:POKE 32059,0
POKE 32058,54:POKE 32059,255
POKE 32058,119:POKE 32059,0

A gépi kód sikeres betöltése után a szolgáltatásait egy példaprogramon ismerhetjük meg. *(2. lista)*

VIGYÁZAT! Hívás előtt az A$ változót mindenképpen töltsük fel; nem lehet üres!

FETSER IGNÁC

```
10 CLEAR 31999:S=0:POKE 38656,125
20 POKE 38655,0:FOR F=32000 TO 32093
30 READ A:POKE F,A:PRINT F,PEEK F
40 S=S+A:NEXT F
50 IF S<>9861 THEN PRINT "ADAT HIBA!"
60 DATA 229,213,197,245,133,128, 80
61 DATA  22,  2, 14,  8,  6, 31,229
62 DATA  35,126, 43,119, 35, 16,249
63 DATA 225, 36, 13, 32,241, 33,160
64 DATA  80, 21, 32,233,229, 42,100
65 DATA 125, 35, 35, 34,100,125, 94
66 DATA  35, 86,122,254, 61, 56, 27
67 DATA 225, 43, 14,  2,  6,  4, 26
68 DATA 119, 36,119,  0, 36, 19, 16
69 DATA 247, 33,191, 60, 13, 32,239
90 DATA 241,193,209,225,255,201, 33
91 DATA 105,125, 34,100,125, 24,205
92 DATA  62,150,237, 71,237, 94,201
93 DATA 237, 70,201
99 PRINT "GEPI KOD BETOLTVE"
```

```
100 LET A$="A KIIRANDO SZOVEG"
110 LET A$=A$+CHR$ 0
120 LET D=32107
130 FOR F=1 TO LEN A$
140 LET A=CODE A$(F)*8+15360
150 LET B=INT (A/256)
160 LET C=256*(A/256-B)
170 POKE D,C
180 POKE D+1,B
190 LET D=D+2
200 NEXT F
210 RANDOMIZE USR 32084:PAUSE 1000
220 RANDOMIZE USR 32091
```

HT Magyarul — hibáztál!

Üzen a gép

A program betöltés után extended üzemmóddal jelentkezik be, és a listában olvasható megfogalmazásban értesíti a gép kezelőjét az előforduló hibákról. A gépben lévő BASIC programot nem törli, és mivel a tár végére íródik, csak kis helyet foglal el a BASIC tárterületből.

```
57          LD    A,'5'
58          CP    (HL)
59          JR    NZ,T1           ;Ugrás ha nem 5-el kezdődik
60 PRZ:     LD    A,'z'           ;"z" kiírása
61          CALL  51
62 T1:      LD    A,32            ;space kiírása
63          CALL  51
64          POP   HL              ;Sorszám(stack) => HL
65          CALL  0FAFH           ;Sorszám kiírása
66          LD    HL,LINE         ;". sorban" szöveg kiírása
67          CALL  2B75H
68          JP    1A19H           ;Beolvasási főágra ugrik
69 ;
70 ; Hosszvizsgálat
71 ;
72 LEN:     LD    D,0             ;Karakterszámláló (D-reg.)
73 T2:      INC   D
74          INC   HL              ;Következő karakterre lép
75          LD    A,(HL)          ;Továbblép,ha még nem érte el a szöveg végét
76          AND   A
77          JR    NZ,T2
78          LD    A,1             ;1-es és 1000-es nagyságrendű
79          CP    D               ;  számoknál a "z" kiírására ugrik
80          JR    Z,PRZ
81          LD    A,4
82          CP    D
83          JR    Z,PRZ
84          JR    T1              ;Egyébként ugrás sorszámkiírásra
85 ;
86 ; Hibaüzenet-táblázat
87 ;
88 BA:      DB    'FOR nélküli NEXT ',0
89          DB    'Formai hiba ',0
90          DB    'GOSUB nélküli RETURN ',0
91          DB    'Elfogytak az adatok ',0
92          DB    'Hibás függvényargumentum ',0
93          DB    'Túlcsordulás ',0
94          DB    'Betelt a tár ',0
95          DB    'Nem létező sorszám ',0
96          DB    'Hibás index ',0
97          DB    'DIM újradefiniálása ',0
98          DB    'Nullával való osztás ',0
99          DB    'Parancsként nem használható utasítás ',0
100         DB    'Változó típuskeveredés ',0
101         DB    'Nincs több hely stringek számára ',0
102         DB    'Túl hosszú string ',0
103         DB    'Túl bonyolult stringművelet ',0
104         DB    'Érvénytelen CONT ',0
105         DB    'Hiányzó RESUME ',0
106         DB    'Hiba nélküli RESUME ',0
107         DB    'Érvénytelen hibakód ',0
108         DB    'Hiányzó operandus ',0
109         DB    'Hibás file-szerkezet ',0
110         DB    'DISC-BASIC utasítás '
111 EA:      DB    0
112 LINE:    DB    '. sorban',13,0
113         END
```

OKTA TOTÓ

Az előző számban TOTÓ jelent meg a számítástechnika történetéből. Most áttekintjük ezt a témakört.

Az ősközösségi számolási „eszköz" az ujj volt. Az állattenyésztés kialakulása kezdetén az állatok megszámolásának első tényleges eszköze a kavics lett. Később az ilyen és hasonló egyszerű esetekben csontokkal, kötélcsomókkal, babszemekkel is számoltak.

A következő említésre méltó tárgy az abakusz, amely már ókori találmány. Ez a keretbe foglalt pálcákra felfűzött golyókból álló segédeszköz a számtan oktatásában sok helyen még ma is használatos.

IGAZI GÉPEK

Wilhelm *Schickard* (1592—1635) 1623-ban épített számológépe, mely 6 helyértékű volt, már a technikai forradalom előfutára. Ettől kezdve érezhetően fölgyorsultak az események.

— Blaise *Pascal* (1623—1663) francia matematikus és természettudós 1642-ben alkotta meg számológépét. A berendezés mechanikus fogaskerékrendszerrel dolgozott, és nyolcjegyű számok összeadására, kivonására volt képes.

Ha kétféle gépünk van, érdemes külön a 16-os és a 64-es gépbe is beírni (a megadott kezdőcímekkel) a programot — a jó tárkihasználás érdekében és az eltérő karakterkészlet miatt. Ha a 64 kbájtos gép programját a 16-osba töltjük, akkor betöltés után „elszáll". Mindkét változat újraállítja a RAMTOP-ot, és így az BASIC-ből nem írható felül.

A programot csak a gép hideginditása hatástalanítja. 64-es gépbe a betöltés után <CTRL> + <1> (<shift> + <clear > + <1>)-et kell írni, mert egyébként nem ír ékezetes betűket.

KATONA GYÖRGY

```
 1 ;****************************
 2 ;*Magyar hibauzenetek HT-ra*
 3 ;****************************
 4         ORG  41A6H          ;Hibakiíró rutin kezdőcíme
 5         JP   START          ;(Hiba esetén ide usrik)
 6         ORG  41E2H          ;Automata indítás címe
 7         JP   INIC           ;(Betöltés után ide usrik)
 8         ORG  40B1H.         ;Ramtop
 9         DW   INIC-1
10         ORG  32070          ;16 kbyte
                64840          ;64 kbyte
11 INIC:  LD   HL,41E2H       ;SYSTEM parancs inicializálása
12         LD   (HL),201
13         JP   12288          ;Extenderd üzemmód
14 START: POP  HL             ;Visszatérési cím torlése
15         LD   A,(403AH)      ;Hibakód => A (0-44)
16         SRL  A              ;Hibakód átalakitása (1-23)
17         INC  A
18         LD   HL,BA          ;Hibauzenet-táblázat kezdőcíme => HL
19         LD   D,A
20         LD   BC,EA-BA+1     ;Táblázat hossza => BC
21 ;
22 ; Hibauzenet keresése
23 ;
24 NEXT:  DEC  C
25         JR   Z,KI           ;Kilépés ha megtalálta
26         XOR  A              ;Keresendő szám(0) => A
27         INC  HL             ;Uzenetet zázó nulla átlépése
28         CPIR                ;Kovetkező szöveszáró nulla keresése
29         JR   NEXT
30 KI:    CALL 2B75H          ;Uzenet kiírása
31         LD   HL,(40EAH)     ;Sorszám(amelyben a hiba fellépett) => HL
32         PUSH HL             ;Sorszám (HL) => "STACK"
33         LD   A,H            ;Sorszám vizsgálata
34         AND  L              ;Parancsban történt hiba esetén
35         INC  A              ;  a beolvasási fóásra usrik
36         JP   Z,1A19H
37         LD   DE,65534
38         RST  18
39         JP   Z,674H
40 ;
41 ;Folytatás programsorban fellépett hiba esetén
42 ;
43         LD   A,'a'          ;"a" kiírása
44         CALL 51
45 ;
46 ;Sorszám átalakítása karakteres formára
47 ;
48         CALL 0A9AH          ;HL => x (egész tipus)
49         XOR  A
50         CALL 1034H
51         OR   (HL)           ;Zero flag torlése
52         CALL 0FD9H          ;x-ben levő számot ASCII-re alakitja
53         INC  HL             ;Kezdő space átlépése
54         LD   A,'1'          ;1-el kezdődő számok esetén hosszvizsgálatra usril
55         CP   (HL)
56         JR   Z,LEN
```

— *Leibniz* (1646—1716) német matematikus és filozófus négyműveletes számológépet szerkesztett 1673-ban, mely a szorzási műveletek végrehajtását automatikusan ismételt összeadással végezte.

— *Babbage* (1792—1871) angol matematikus 1822-ben készített egy berendezést, amellyel táblázatokat lehetett kiszámítani. Babbage-nek 1812-ben régi számításokat kellett ellenőriznie, amelyek tele voltak hibával. A feladatot egyedül nem tudta elvégezni, ezért a munkához Prony módszerét alkalmazta: a feladatot elemeire bontotta. Három munkacsoportot szervezett, úgymint:

— képzett matematikusok,
— gyakorlott számolók,
— számológépek (nekik csak összeadniuk és kivonniuk kellett).

Ezt a munkamegosztást úgy képzelhetjük el, mint a mai feldolgozás alábbi funkcionális megfeleltetését:

— rendszerelemzők (szervezők),
— programozók,
— számítógép.

(A rendszerelemzők által megállapított módszert a programozók készítik elő a számítógép számára.)

Korszakalkotó találmányok

Új korszakot egy „apróság" nyit: a lyukkártya. Feltalálása *Falcon* nevéhez fűződik, akinek 1728-ban lyukkártyás vezérléssel automatizált szövési eljárását jóval később Jacquard tökéletesítette (1801-ben). 1812-re több mint 11 000 Jacquard gép üzemelt Franciaországban. A gép bal oldalán lyukkártyák voltak, melyek az olvasószerkezet alá érve, az érzékelők közvetítésével kapcsolatba kerültek a szövőkarokkal, és így a géppel bonyolult mintákat lehetett készíteni. (A magyar szakirodalom sajnos elhallgatja Falcon szerepét.)

Hermann *Hollerith* (1860—1939) használta a lyukkártyát elsőnek adathordozóként statisztikai munkára. 1890-ben az USA-ban lyukkártya segítségével értékelték ki a népszámlálási adatokat; a lyukkártya szabványosított mérete az akkori egydollárosnak felelt meg. (A lyukkártyás adatfeldolgozás általános elterjedése 1925-től vehető.) A rendezőgép a lyukkártyák különböző szempontok szerinti sorba rakását segítette elő. A rendezett kártyák információinak kinyomtatását az ún. táblázógéppel végezték.

SZÁMÍTÓGÉP-GENERÁCIÓK

Ezek a gépek külső programozásúak voltak, azaz kapcsolótáblán, kap-

DIÁK C64 ROVAT

Autostart kazettára

A program Ø. sorába a REM után egy szóközt és 30 darab A betűt kell írni. A futtatás után keletkezett Ø. sorhoz hozzáfüzzük a saját programunkat, majd kimentés előtt beírjuk:

```
POKE 53265,11:POKE 788,7:POKE
789,8:
POKE 808,254:POKE 44,3:CLR:
SAVE"NÉV",1,1:RETURN
```

Az így kimentett program rögtön elindul behívás után, és még a RUN-STOP vagy RESTORE billentyűkkel sem állítható le. Ezzel elérjük, hogy mások nem listázhatják és nem javíthatják át. Ha azt akarjuk, hogy a program megállítható legyen, akkor kimentés előtt nem kell beírni a POKE 808,254-et.

```
Ø REM AAAAAAAAAAAAAAAAA
  AAAAAAAAAAA
1Ø DATA 173,17,208,41,16,208,3,76,49,
  234,169
2Ø DATA 49,141,20,3,169,234,141,21,3,
  169
3Ø DATA 8,133,44,32,113,168,76,49,234
4Ø FOR A=2Ø55 TO 2Ø84:READ Q:
  POKE A,Q:S=S+Q:NEXT
5Ø IFS<>3ØØ1 THEN PRINT "HIBÁS
  DATA . .!":END
6Ø PRINT " ♥QQ":FOR A=1Ø TO 7Ø
  STEP 1Ø:PRINT A:NEXT:PRINT
  "☜"
7Ø FOR A=631 TO 639:POKE
  A,13:NEXT:POKE 198,9
```

RÓSA ÁDÁM

csolózsinórok dugaszolásával kellett működtetni őket. A táblázógép jellemző áramköri eleme a relé volt. Az e géppel, illetőleg alkatrésszel képviselt technikai szintet a számítástechnikában a *nulladik generációnak* szokták mondani.

Az *1. generációs számítógép* jellegzetes áramköri eleme az elektroncső volt. Az ENIAC 1945 decemberében készült el és kb. 18 000 db elektroncsővel dolgozott. (Érdekességként a gép méretei: 30,5 méter hosszú, 3 méter magas és egy méter széles.) Ez a szintén külső programozású gép kevesebbet tudott, mint a mai legegyszerűbb személyi számítógép!

A *2. generációs számítógépek* (1958-tól terjedtek el) már tranzisztort és diódát is tartalmaztak. A rájuk még nem igazán jellemző belső programozás elve *Neumann János* (1903—1957) nevéhez fűződik (munkásságát az 1987/4. számunk részletesen méltatja — a szerk.). Az ő ötlete volt, hogy utasításkódként is számjegyeket alkalmazzanak, amelyeket elektronikusan lehet tárolni. Így a számítógép — ha szükséges — képes saját programját is módosítani. A program így a számítógép belső tárába kerülhetett. Neumann nyomán már 2. generációs gépet is készítettek, ami belső programozású volt.

A *3. generáció* korát 1965-től számítják. Ezek a gépek integrált áramkörökből (IC) épülnek fel. Egy IC sok tranzisztort és egyéb áramköri elemet foglal magába, de a terjedelme csak néhány mm².

A fölzárkózás kezdetei

1969-ben megismerkedhettünk az ESZR betűcsoporttal, melynek jelentése: Egységes Számítógépes Rendszer. Ezt a gépcsaládot a szocialista országokban a KGST-ben egyeztetett koncepció szerint készült, a kor technológiai színvonala által meghatározott számítógépek alkotják. Ide tartozik például az R20-as gép, amely a gépcsalád első sorozatú berendezése (típusszáma 0-ra végződik; az R35-ös például már a kettes sorozat gépe). A koncepció lényege — és a gépcsalád tagjainak rendszerjellemzője —, hogy az újabb, modernebb gépen a régi programok változtatás nélkül futtathatók.

Ezen a ponton érdemes kitérni a robottechnikára. Az Egyesült Államokban 1961-ben adták el az első robotot; Japánban 1967-ben helyezték üzembe az elsőt. Magyarország 1972-ben szerzett be egy szervovezérléses robotot; 1985-ben alakult meg a Magyar Robottechnikai Társaság. 1986-ban 60-80 robot volt az országban.

A mikrogépek forradalma

A hazai szakirodalomban tévesen az terjedt el, hogy az ősmikroprocesszort az Intel vállalat 1971-ben készítette el. A valóságban a Texas cégé az ezzel kapcsolatos úttörő — szabadalmaztatott — megoldás. Az első, sériában gyártott személyi számítógépnek mondható berendezést 1975-ben jelentette meg a MITS cég, Altair 8800 néven. Az első, viszonylag olcsó, nagy sorozatú mikrogép a Commodore cég PET-je volt. Az eddigi legsikeresebb, amatőrök által kifejlesztett gép az Apple II számítógép. Előállítója Steve *Wozniak* és Steven *Jobs*.

A személyi számítógépek kategóriájában fontos állomást jelentett az 1980 januárjában megjelent ZX80-as gép, melyet Sir Clive *Sinclair* állított elő. A Sinclair cég ezt a gépet a többi hasonló kategóriájú géphez képest olcsón dobta piacra. A márka sikerét fokozta, hogy 1981-ben a ZX81-et, majd 1982-ben a ZX-Spectrumot is forgalomba hozták, és ezek a fejlesztések egymásra épültek. 1982-ben jelent meg a Commodore 64-es számítógép is. (Eddig milliós példányszámban eladott gép az Apple II, a C64, az IBM PC és XT, a Tandy Coco, a Texas TI 99/4A, a VC20 és a ZX81.)

Léptékváltás

Közben az ESZR mintájára megalakult az MSZR (1977-ben), a Mikroszámítógép Rendszer, amelyet szintén a szocialista országokban gyártott, egyeztetett követelményű és kategóriájú gépek — nem személyi számítógépek! — alkotnak.

Magyarország — és egyben a szocialista országok — legelső sorozatban gyártott háziszámítógépe az Aircomp—16 volt. Ez a típus szintén 1982-ben került piacra. A hazai iskolaszámítógépek során pedig a HT—1080Z nyitotta meg 1983-ban; képességei megfelelnek az USA-ban 1977-ben gyártott TRS-80 egy egyszerűsített változatának. 1984-ben jelent meg a Primo, amelyet az idén februárban Budapesten már 2500 forintért árultak. A Videoton TV-Computere 1986-ban került a vásárlókhoz.

Egy rövid de tanulságos történettel szeretnénk most megszakítani a szárazzá váló krónikát. 1986-ban a Sinclair céget el kellett adni. A csőd egyik okozója az újabb számítógép, a QL volt. Bár ez sokat tudó, jó gép, mégis lerúgta az égről a Sinclair-csillagot. Hogy lehet ez? Nos, mivel több — módosított — verziója került forgalomba, a programokat gyártó cégek — nem tudva vagy nem akarva a szexélyes piacon kockáztatni — elpártoltak ettől a géptől. A Sinclair márkanévvel együtt az Amstrad cég vásárolta meg a maradványokat, és 1986-ban már ki is hozta a Spectrum 128 k+ 2-es gépet.

Érdekes módon a Commodore cég is a csőd szélére került; a C64 már kifutott, a C16 rossz üzlet volt, és a C+ 4 sem hozta a várt sikert. Az Egyesült Államok külön pénzt folyósított C-éknek, abban a reményben, hogy az új típusú számítógépük, az Amiga talpra fogja állítani a céget. Egyelőre azonban az Amiga még nagyon drága.

Az IBM— melynek egyik alapítója (tulajdonképpen jogelődjének alapítója) maga Hollerith volt — közben kihozta az IBM PC-ket. Az ár miatt egyelőre még nem mondható, hogy ez nálunk otthoni személyi számítógép tipus lehet, de több cég is törekszik rá, hogy az IBM PC-vel kompatibilis számítógépet gyártson.

Az előzőekben eljutottunk a 3. generációs számítógépekig. Érdemes tovább vizsgálni ezt a vonalat. A *4. generációról* — melynek megjelenését az 1980. évtől említik — két felfogásban is szólhatunk. Az egyik szerint — csakúgy, mint a 3. generációs gép — integrált áramkörökből épül fel, csakhogy ezek már magasabb fokon integráltak. Ebben a verzióban a generáció jellemzője még, hogy ez a technológia és a hozzá kapcsolódó hardverszoftver támogatja a számítógép-hálózatokat. A másik nézet lényege, hogy a mágnesbuborékos tárolót tekinti a 4. generációt jellemző eszköznek. (Egyébként az egyik értelmezés nem zárja ki a másikat! — A szerző.)

Az *5. generációs számítógépek* megalkotását 1990-re tervezik. Ezek ún. mesterséges intelligenciával rendelkező, igen nagy integráltságú gépek lesznek. (Olvasóink ezekben a hónapokban találkoznak sorozatunkkal, mely a tárgykört részletezi. — A szerk.) Jellemzőjük a közvetlen ember-gép kapcsolat, a logikai programozás és az ún. párhuzamos áramkörök. Fontos szerepet játszik a feldolgozás sebességének növekedése is. (A téma iránt mélyebben érdeklődőknek ajánljuk: H. Goldstine: A számítógép Pascaltól Neumannig. Budapest, 1987. Műszaki Könyvkiadó, 368 old., 110,— Ft. — A szerk.)

Az 1987/5. számunkban közölt TOTÓ helyes megoldása:
2 x 2 1 2 x 1 2 2 2 x 1 x 1

OKTA-TOTO

A második forduló témaköre a hardver. A megfejtéseket a pályázati szelvényre kell rávezetni úgy, hogy a kérdésszám sorába a játékos beírja tippjét: 1-est, 2-est vagy X-et. A közölt három válasz közül csak egy helyes.

A kitöltött szelvényt postai levelezőlapra felragasztva a szerkesztőség címére küldjék be (1371 Budapest, Pf.: 433).

A beküldési határidő: 1987. június 20.

A helyes megoldás feladói közül havonta öt nyertest sorsolunk ki, akik 200 forint értékű könyvutalványt kapnak. Akinek mind a hat TOTÓ-ja telitalálatos, számítógépet nyerhet!

A pályázati szelvény nélkül érkezett megoldás nem vesz részt a sorsoláson!

1. Egy dobrendszerű sornyomtató
1. egy nyomtatási ciklus alatt egy sort nyomtat
2. egy nyomtatási ciklus alatt egy karaktert nyomtat
X. egy nyomtatási ciklus alatt egy lapot nyomtat

2. A mágneslemezekre a soros adatállományok felírása
1. szektoronként véletlenszerűen
2. lemezoldalanként sorban
X. cilinderenként sorban történik

3. A mágnesszalagokon a szalag elején minden esetben található
1. szalagkezdet-jel
2. állománykatalógus
X. tisztító-befűző szalagdarab

4. Melyik a része a központi egységnek?
1. vezérlőegység
2. aritmetikai egység
X. mindkettő

5. Az elektronikus /félvezető elemeket tartalmazó/ operatív tár
1. sebessége kisebb a ferrittárénál
2. hálózatkimaradás esetén törlődik
X. szervezését a gyártási technológia határozza meg

6. A vezérlőegység utasításregisztere
1. a futó utasítás címét tartalmazza
2. az éppen végrehajtás alatt álló utasítást tartalmazza
X. a műveleti kódot tartalmazza

7. A mikroutasítás
1. egy mikroprogram egyetlen utasítása
2. egy mikroprocesszor-utasítás
X. a gépi utasítások egyik része

8. Egy C64 alapgép
1. jóval olcsóbb, mint egy színes televízió
2. kb. ugyanannyiba kerül, mint egy színes televízió
X. jóval drágább, mint egy színes televízió

9. Egy személyi számítógéprendszerben
1. a botkormány és a képernyő közvetlenül össze van kötve
2. a botkormány és a képernyő is a központi egységhez csatlakozik
X. a központi egységhez egyszerre nem csatlakozhat a botkormány és a képernyő

10. Mi a megszakításregiszter tartalma?
1. a megszakítást okozó esemény által küldött információ
2. a megszakított programot jellemző információ
X. információ a megszakításkérésekről

11. Mi az a prioritás?
1. ütemezés
2. elsőbbség
X. kiválasztás

12. Ha egy személyi számítógéprendszerben két mágneslemezes egység van, akkor ezeknek
1. a fizikai egységszáma megegyezhet, de a logikai egységszámuk mindig különböző
2. a logikai egységszámuk megegyezhet, de a fizikai egységszámuk mindig különböző
X. mindkét egységszámuk mindig különböző

13. Egy mikroprocesszor címvonalai a legtöbb esetben
1. egyirányúak
2. kétirányúak
X. a működtető programtól függően egy vagy kétirányúak

+1. Egycímes utasítás esetén
1. a művelet csak egy operandussal zajlik
2. a művelet két operandussal zajlik, és a második operandus helyét a processzor határozza meg
X. a művelet tetszőlegesen sok operandussal végtelenített

PÁLYÁZATI SZELVÉNY 87/6

OKTA-TOTO

Kérdés	Tipp
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
12.	
13.	
+	
13+1.	

Név: _____

Cím: _____

_____

Szem. szám:

HÁZI- ÉS ISKOLASZÁMÍTÓGÉP-MÉRLEG

Háziszámítógépek

Tavaly a hazai háziszámítógép-állomány mintegy 20 ezerrel bővült. Ez abszolút értékét tekintve magasabb ugyan, mint az 1985. évi növekmény, százalékban azonban „csak" 40%-os növekedést figyelhetünk meg. Ez érezhető mérséklődést jelent az 1982—85 között megszokott 2-3-szoros növekedési ütemhez képest.

A *típusválasztéka* ennek a kategóriának 1986-ban már csak egyetlen új, általános célú géppel, a HT 3080 C-vel bővült. Ez önmagában is mutatja, hogy ennek a teljesítménykategóriának az életciklusa már a lefelé menő ágában van.

A *hazai gyártású* gépeknek az 1985-ben már ismert típusválasztéka szerepelt a hazai kínálati palettán, szinte változatlan összetételben. Az elmúlt év legfontosabb eseménye a háziszámítógép-piacon a Videoton TV Computerének a boltokban való megjelenése. Bár a gépet magát már 1984-ben bemutatták, végül 2 esztendőre rá jelent csak meg a forgalomban. Az időpont nem éppen szerencsés, hisz ez a teljesítménykategória már kifutóban van, a mikroszámítógép-divat is kezd lecsengetni s így a piacon kialakultak a stabil típusok. Mindezt nyilvánvalóan érzi a Videoton is, ezért is igyekszik gazdag perifériaválasztékot (nyomtató mellett hajlékonylemezes tárolót is) biztosítani a gépének, ad hozzá továbbá CP/M operációs rendszeropciót (UPM-nek nevezte), s már az induláskor terminálfunkciókkal is felruházta. Mindez bővíti az alkalmazási kört s így a felvevőpiacot. A megjelenés előtti árcsökkentés is ennek jegyében történt: az 1985-ben még 19 900 forintért beharangozott gépet végül a boltok 12 800 forintért kezdték árusítani.

1986 szeptemberének szomorú szenzációja a Primo halála. 1984 tavaszán jelentették be az elkészültét, s az akkori tavaszi BNV-n próbálhattuk ki az első gépeket. A csak fekete-fehér lehetőségeket biztosító Primónál a célzott terület a lakosság mellett az általános iskolák voltak, hisz ott általában még csak fekete-fehér televízió van. 1985 novemberében bemutatták a színkezelési lehetőségeket is tartalmazó, mozgó billentyűzetű Pro/Primo-t. Még a Software '86 kiállításon 28 ezer forintos várható árat emlegettek, tavasszal a boltokban ez az ár már csak 21 ezer forint lett. 1986 tavaszán az iskolaszámítógép-pályázaton a Pro/Primónak sem sikerült kizárólagos iskolaszámítógéppé válnia, ami továbbra is bizonytalanná tette a piacát (az iskolaszámítógép-pályázat értékelését lásd később). 1986. július 1-jétől még a forgalom növelése érdekében utolsó próbálkozásként kb. 30%-kal mérsékelték az árát. Mivel ez sem eredményezte az értékesítés számottevő növekedését, augusztusban döntés született a gyártás leállításáról és szeptember 1-jétől elkezdődött a készletek végkiárusítása.

A Primo 30 hónapos életét értékelve elismeréssel kell szólni a kezdeményezés úttörő jellegéről (a szocialista országok első nagyobb sorozatban gyártott háziszámítógépe volt), a kitűnő piaci munkáról (a szocialista országok első, boltban árusított gépe címet is elnyerte). Az élete rövidségéért okolható az elterjedt nyugati típusokkal való inkompatibilitása, az elég megbízhatatlan program ki- és bemenete, a mindezek következtében gyenge programellátottsága. Ez utóbbi a programok ezreivel kényeztetett Commodore 64 és Spectrum gépekkel szemben óriási hátrányt jelentett. Magyarul: nem lehet ma már csupasz vasat értékesíteni.

A biztonsági szelepet a szocialista piac jelenthette volna a gép számára. Azonban ez a gépkategória csak a lakossági fogyasztási cikkek bővülését eredményezné, s a termelőeszköz-orientált országokban ez nem elég vonzó. Így a Primo, körbejárva a környező országok vásárait, egyetlen külpiacra sem tudott betörni.

További 1986. évi újdonság, hogy a Homelab-gépek is megjelentek januártól a kiskereskedelmi forgalomban.

Ebbe a kategóriába tartozó új hazai gép a vakok számára a Brailab. Ez a legépelt betűket rögtön számítógépes úton szintetizálva hangosan ki is mondja. A Tudományszervezési és Informatikai Intézet 1986-ban 23 darabot vásárolt belőle s kiosztotta a Vakok Általános Iskolájának, a Vakok Állami Intézetének, s kapott egy-egyet az ELTE (szofverfejlesztésre) s a Gyógypedagógiai Főiskola (tanárképzésre). A gépet, mint a neve is mutatja, a Lukács-testvérek a Homelab nevű gépükből alakították ki, a hangszintetizáló részt pedig az MTA KFKI-ban fejlesztették (Arató András és Vaspöri Teréz).

Az *utasforgalomban* behozott típusok között továbbra is a Commodore 64 vezet. Nem sokkal marad el tőle a Sinclair gépek száma. Ezek összetétele viszont jelentősen megváltozott. Szinte már senki sem hozott be ZX81-et, csökkent a vásárolt ZX—Spectrumok száma, viszont jelentősen nőtt a ZX—Spectrum Plus népszerűsége.

A Commodore 16-ok behozatala leapadt, hisz leállították a gyártását. Helyette a vele kompatibilis, hasonló árfekvésű Commodore Plus 4 száma növekedett jelentősen.

Nem érzékelhető a Commodore 116, 128, a ZX—Spectrum 128 hazai számának számottevő növekedése, ezek felfutása már nem is várható.

A típusválaszték tekintetében megállapíthatjuk, hogy 1986-ra már egészen polarizálódott a háziszámítógép-piac. A hazai vásárlók túlnyomó többsége a Commodore 64 és a ZX—Spectrum, valamint az azzal kompatibilis ZX—Spectrum Plus típust választotta. E döntést alapvetően az e két típushoz rendelkezésre álló gazdag, ezres nagyságrendű szoftverkészlet motiválja. A háziszámítógépek nyugati típusválaszték is kialakult, újabb jelentős típusok megjelenése már nem várható. Ugyanez elmondható a szoftverkínálatra is: a választék jelentős bővülésére már nem lehet számítani.

Az 1986-os év fordulópontot jelentett a Commodore 64 hazai megítélésében: csak elvétve vásároltak belőle professzionális célokra. Ez elsősorban a 16 bites gépek hazai árának a csökkenésével magyarázható.

A mikroszámítógépeket *forgalmazók köre* 1986-ban jelentősen bővült. A hagyományos bizományosi hálózaton túlmenően az esztendő a hazai gyártású gépek kínálatának bővülésével indult. Mint ismeretes, a Primo gépet már 1985 tavaszától elkezdték a Keravillban árusítani. 1986 januárjától felzárkózott mellé a Homelab 4, majd tavasztól a Videoton TV Computere.

Az utóbbit a Centrum Áruházak vették át országos forgalmazásra, ezáltal a TV Computer volt sok vidéki városban az első, boltban kapható számítógép. Ennek a számítástechnikai kultúra terjesztése szempontjából nem elhanyagolható a hatása.

A háziszámítógépeket forgalmazók köre 1987-ben várhatóan tovább bővül, hiszen messze van még a fejlett országokban ismert mennyezettől: ott már évek óta a trafikokban is kaphatók háziszámítógépek.

Az *árak* csökkenése 1986-ban folytatódott. A nyugati gépek hazai árai az év során mintegy 20-30%-kal mérséklődtek, azaz lényegében követték a kinti árcsökkenést. Ez a követés azonban abszolút összegét tekintve elég távoli. Egy Commodore 64 az év végén az NSZK-ban 400 márka (kb. 10 ezer forint) körüli áron szerezhető be, a hazai bizományi ára pedig ugyanakkor éppen csak 30 ezer forint alatt van. Ebben a magas árban feltétlen szerepet játszik a gép 18

Kategória	Helyezés	Típus	Tár (Kbájt)	Gyártó/forgalmazó	Nagyker. ár (Ft)
Általános iskola	1-2	Commodore 16	64	Novotrade/Piért	8 775
	1-2	TV Computer	32	Videoton	11 000
	3	Pro/Primo	32	MTA SZTAKI-Microkey	13 500
Közép-iskola	1	Pro/Primo	64	MTA SZTAKI-Microkey	17 000
	2	TV Computer	64	Videoton	13 000
	3	HT 3080C		HTSZ	28 000

ezer forinton „felejtett" vámértéke, mely így a kinti árnak csaknem kétszerese, s a 30% vámmal a vásárlók lényegében a kinti vételár felének megfelelő vámot fizetnek.

A hazai gyártású gépek ára is 30-70%-kal mérséklődött. A legjelentősebb ezek közül a Primo 1986. évi kétszeri árcsökkentése, melynek eredményeképpen az év végére a gép ára a harmadára csökkent.

Az elmondottakból világosan következik, hogy háziszámítógépekből 1986-ra a hazai piacon a túlkínálat jelei mutatkoztak, a vevők válogathattak a különféle típusok között.

Iskolaszámítógépek

Ez a kategória erősen összefügg a háziszámítógépekkel, ezért csak az azoktól eltérő dolgokra térünk ki.

A tavalyi évnek e kategória tekintetében a legnagyobb eseménye a második hazai iskolaszámítógép-pályázat tavasszal tartott eredményhirdetése volt.

Az egyetlen külföldi típust, a Commodore 16-ot — melyet a Novotrade importált — csak 1986-ban árusították, mivel már megszűnt a gyártása. A magyar típusok vonatkozásában a pályázat eredményhirdetése alapvetően különbözött az 1982. évitől. Akkor egyetlen típust választottak ki, a HT 1080Z-t, s abból nagybani vásárlást garantáltak. Ez monopolhelyzethez juttatta a gyártót, ami bizonyára érződött mind a minőségben, mind pedig a rendkívül — és sokáig — magas árban. Ez utóbbi még talán a fenti táblázatban is kísért, hisz a ZX—Spectrum kompatibilis HT 3080C gépet nagybani szállításra a nyugati kiskereskedelmi ár csaknem ötszörösével, a hazai bizományosi árnál 50%-kal magasabban, s hasonló paraméterű konkurens ajánlatoknál 50-100%-kal magasabb árért ajánlották meg.

A helyezett típusok közül az iskolák szabadon választhatnak, s ebben fontos szerephez jut az ár. A táblázatbeli árak már nem aktuálisak, s az iskolák kedvezményt is kapnak. Bővebb és pontos információt a Tudományszervezési és Informatikai Intézet ad.

Az 1986. évi iskolaszámítógép-pályázat eredményhirdetése által kitermelt konkurencia első áldozata a Primo lett: 1986 augusztusától megszüntették a gyártását. A gyorsan változó világot jól jellemzi, hogy alig 6 hónappal az eredményhirdetés után a 6 tételes eredménytáblázat már megváltozott: a Pro/Primo kiesett, a Commodore 16 pedig átalakult Commodore Plus/4-gyé.

Az iskolai gépbeszerzések volumene komoly. Az iskolaszámítógép-programnak az 1986 tavaszán indult második lépcsője 1990-ig 50 ezer iskolaszámítógép beszerzését irányozza elő, ezen fele-fele arányban osztozik a két iskolatípus.

Az új iskolaszámítógép-program igen lendületesen indult: 1986-ban megnégyszereződött az iskolaszámítógépek száma. 1985 végén az általános és a középiskolákban még csak háromezer-hatszázhatvannyolc számítógép szolgálta a tanuló ifjúságot, 1986 végére pedig már 15 ezer. Ez egyben azt is jelenti, hogy míg 1985-ben több mint négyszáz diákra jutott egy számítógép, tavaly viszont már alig több, mint kilencvenre. A rendkívüli ütemű növekedésben jelentős szerepet játszott a ráckevei Aranykalász Mezőgazdasági Termelőszövetkezet által kezdeményezett „Számítógépet az iskoláknak!" mozgalom: 1986 elején az ország valamennyi gazdálkodó szervezetéhez elküldte felhívását. Ez széles körű visszhangra talált. Több száz intézmény, vállalat és termelőszövetkezet, budapesti kerületi és megyei tanács vásárolt a vonzáskörzetükben működő iskolák számára számítógépet. Például a XIII. kerület iskoláinak az úttörőszövetség megalakulásának 40. évfordulójára 204 mikroszámítógépet ajándékoztak. Az állam a „Számítógépet az iskoláknak!" mozgalmat 33%-os dotációval támogatja.

Az ilyen arányú gépbeszerzésiigény-növekedés kielégítésének a megszervezése szintén nem mindennapi feladat volt. A Tudományszervezési és Informatikai Intézet ebben hatékony partnerre talált a Novotrade cégben. Ez a nyáron 6500 db Commodore Plus/4 típusú gépet hozott be, melyek a tanév elején iskolákba kerültek 4990 forintos áron. Ezen kívül 2600 forintért minden iskola kapott egy egységcsomagot, amely magyar nyelvű szakirodalmat és két programkazettát is tartalmazott. A gépek eleve magyar karakterkészlettel kerültek forgalomba. A gyerekek számára segítségképpen elkészült egy sokszínű, műanyag lapokból álló oktatótábla-sorozat is. A Plus/4-hez már szeptemberben is kapható volt hatféle magyar nyelvű szakkönyv. Köztük megjelent a sikeres Hetedhét sorozat legújabb tagja, a Hetedhét Plus/4, valamint a Bevezetés a Basic-be c. könyv oktatókazettával együtt. A Plus/4-et oktató pedagógusok felkészítését is gondosan szervezték meg. A Novotrade e téren szövetkezett a számítástechnika-oktatás terén nagy hagyományokkal rendelkező Számalkkal, s közösen szervezték meg a tanárok számára tartott továbbképző tanfolyamokat.

Összefoglalva megállapíthatjuk, hogy

— a háziszámítógépekből 1986-tól a hazai piacon a túlkínálat jelei mutatkoznak, a vevők válogathatnak a különféle típusok között;

— míg a háziszámítógép-állomány gyarapodási üteme 1986-ban mérséklődött, az iskolaszámítógép-állományé viszont jelentősen megnőtt. A két kategóriában az év során együttesen mintegy 30 ezerrel bővült az 1985 végén csaknem 50 ezerre tehető hazai állomány;

— a háziszámítógépek nyugati típusválasztéka kialakult, újabb jelentős típusok megjelenése ebben a teljesítménykategóriában már nem várható;

— a nyugati gyártmányú háziszámítógépek szoftverkínálata kialakult, a választék jelentős bővülésére nem lehet számítani;

— a háziszámítógépek hazai és nemzetközi árai kialakultak s viszonylag stabilizálódtak.

DR. BROCZKÓ PÉTER

Jellemző házi- és iskolaszámítógép-árak

(-: az adott típust még vagy már nem gyártják/forgalmazzák)

A számítógép típusa	Gyártó/forgalmazó	Ár (1000 Ft)						
		1983	1984	1985			1986	
		december	december	június	december	március	június	december
Commodore 64	BÁV	100	90	49	49	35	35	30
	Fotoelektronik	91					31	25
	Ofotért	66					30	24
	Ramovill	88				33	33	30
Commodore 128	BÁV	-	-			68	65	65
	Fotoelektronik	-	-				68	49
	Használtcikk Szöv.	-	-				68	49
	Ofotért	-	-			68	66	49
Commodore VIC 20	Ramovill	45				30	25	
HT 1080Z	HTSZ	58	-	35	35	35	35	17
Homelab 4	Color Ipari Szövetkezet	-	-				15	15
Primo 32K	EMO	-	15	15	15	9	9	5
Primo 48K	EMO	-			20	12	12	7
64K	EMO	-			24	14	14	8
Pro/Primo	EMO	-			28	21	21	12
PTA 4000	HTSZ	-	16	13	12	12	12	12
TV Computer	Videoton	-	-	-	20	12	12	12
ZX Spectrum 16K	BÁV	50	25					15
48K	BÁV	70	40	30	25	20	20	20
	Fotoelektronik	52					20	17
	Ofotért	49					17	
ZX81 (1K)	BÁV	16	16				6	6

ATARI kontra COMMODORE 64

Általános összehasonlítás

Magyarországon az utóbbi másfél évben a C64-gyel azonos kategóriájú ATARI ugyancsak népszerűvé vált. Mindkettő a személyi számítógépek (home computer) családjába tartozik, de működésükben kisebb-nagyobb eltérések tapasztalhatók.

Cikkünkben irodalmi adatok és tapasztalataink alapján rendszerezzük ezeket az eltéréseket. Ehhez a hardver ismertetéséből indulunk ki, majd táblázatokban állítjuk egymás mellé a legfontosabb jellemzőket. Ahol az azonosságok többségben vannak, csak a különbségekről esik szó (ld. a BASIC interpreter kulcsszavait). Bízunk benne, hogy az olvasó cikkünk információi alapján könnyebben dönti majd el, hogy konkrét feladataihoz azonos hozzáférési lehetőségeket feltételezve melyik géptípust válassza *(1. táblázat)*.

A továbbiakban megvizsgáljuk a leggyakrabban használt egységek fontosabb jellemzőit *(2. táblázat)*. Az egyszerűség kedvéért a C64 jelöléshez hasonlóan az ATARI 800 XL alapgépet is rövidítve, A—800 XL-nek fogjuk hívni.

AZ ALAPGÉP

Az A—800 XL 16-féle grafikus üzemmódot ismer, melyből az első három (0—2) text (szöveg) üzemmód, a többi grafikai. A számítógép bekapcsolásakor a 0. üzemmód működik, a további üzemmódokat a GRAPHICS utasítással válthatjuk ki. A grafikus üzemmódban 128 szín, illetve színárnyalat áll rendelkezésre, ami a 8. üzemmódtól a duplájára is növekedhet. Ezzel szemben a C64-en a VIC II chip hétféle grafikus üzemmódjából négyben 16-féle színnel gazdálkodhatunk.

A gép soronként, azonnal interpretál, ami a C64-hez képest szokatlan, de feltétlenül üdvös jelenség. Ugyanis a C64-nél, ha egy utasítássort hibásan gépeltünk be, attól a memória még elfogadja, és csak később, a futtatás során derül fény a szintaktikus hibára, hogy ti. elgépeltünk valamit.

1. táblázat

FELEPITéS, KONFIGURACIO

EGYSéG	ATARI	COMMODORE
Alapgép	800 XL	C=64
Lemezmeghajtó	1050	1541
Nyomtató	1029	MPS 801 vagy egyéb CBM típus
Képernyő	TV vagy monitor	TV vagy monitor
Plotter	1020	1520
Magnetofon	1010	1530
RS 232 csatorna	igen	igen
Botkormány (joystick)	2 db	2 db
Cartridge	igen	igen
Fényceruza	igen	igen
Rajzolótábla	igen	igen
Memóriabővítő	igen	igen

Az A—800 XL-nek öntesztelő rendszere van, amely a BYE parancs hatására a képernyőre vetíti a lehetséges tesztváltozatokat; így:

SELF TEST
- MEMORY — memóriateszt
- AUDIO-VISUAL — hangteszt
- KEYBOARD — a billentyűzet tesztelése
- ALL TEST — az előző hármat egyszerre teszteli

Ugyanez ilyen formában a C64-en nincs meg. Az öntesztelő rendszerből a RESET-tel lehet kilépni.

BILLENTYŰZET

Az alapgép billentyűzetén található alfanumerikus jelek elhelyezése megfelel a megszokott elrendezésnek (amerikai billentyűzet), csupán a funkcióbillentyűk helye és szerepe tér el. A *3. táblázatban* a funkcióbillentyűk használatát írjuk le az ATARI szemszögéből.

Az ATARI alapgép és a perifériák között az átvitel szerepét nyolc csatorna látja el (0—7), amelyekből három csatorna (0, 6, 7) foglalt. Ezek a grafikus üzemmódokban használatosak, mégpedig a képernyő alján lévő ablakban történő kiíráshoz. A C64 16 csatornával rendelkezik a következő bontásban:

SAVE	0
LOAD	1
Adatcsatorna	2—14
Parancs- vagy hibacsatorna	15

2. táblázat

ALAPGEP

	A 800 XL	COMMODORE
RAM	64 Kbájt	64 Kbájt
Felhasználói terület	37902 bájt *	38911 bájt
Központi egység	MOS 6502	MOS 6510
ROM	20 Kbájt	20 Kbájt
Programértelmezés	BASIC interpreter	BASIC interpreter
Legmagasabb címzés	65535	65535
Kurzor	merev	villogó
Képernyőszerkesztés	teljes képernyős	teljes képernyős
A képernyő mérete	40 * 24 **	40 * 24
Átviteli csatornák száma	8	16
Grafikus üzemmód	16	7
Max. szín, ill. árnyalat	128, ill. 256	16
Hangcsatorna	4	4

* A DOS bevitele után ez ténylegesen csak 32274 bájt. A C=64-nél a DOS ROM-ban van, és az 1541-es meghajtóban található.

** A képernyő ugyan 40 * 24 karakteres, de a 40 karakterből a szövegszerkesztő (text módban) csak 38 karaktert használ fel, a 0. és az 1. oszlopra külön kell pozicionálni. Ezzel szemben a C=64 nél a 0. oszlopról indul a szerkesztés és a képernyő 25 sorból áll.

BILLENTYŰ	ATARI 800 XL	COMMODORE 64
CRSR	Négy billentyű a CTRL billentyűvel vezérelve; ciklikusan dolgozik, nem léptet sort.	Két billentyű szolgál a kurzor mozgatására. A fel és a balra haladást a SHIFT vezérli.
CLEAR	Képernyőtörlés; SHIFT-tel és CTRL-lal vezérelve ugyanez.	SHIFT-tel vezérelt CLR/HOME.
INS	CTRL-lal vezérelve 1 karaktarnek készit helyet; SHIFT-tel vezérelve egy sornak készit helyet.	INST/DEL SHIFT-tel vezérelve egy karakternek készit helyet, egyuttal a kurzortól jobbra lévő sort jobbra tolja.
DELETE/ BACK SPACE	SHIFT-tel vezérelve egy utasitássort töröl; CTRL-lal vezérelve a kurzor helyén lévő karaktert törli és a tőle jobbra lévőket illeszti a helyébe; önmagában: BACK SPACE: a kurzort balra lépteti és az ott lévő karaktert törli.	INST/DEL a kurzort balra lépteti, az ott lévő karaktereket törli, egyúttal a kurzortól jobbra lévő sort balra huzza.
BREAK	Programkészitéskor megszakitja a programsort és a következő sor elejére lép (nem RETURN!); sort léptet megszakitja a program futását.	RUN/STOP Megszakitja a program futását; SHIFT-tel használva betöltéskor azonnal inditja a betöltött programot.
RETURN	Az utasitássort a memóriába vitele előtt, átadja az interpreternek ellenőrzésre, majd az utasitássort beviszi a BASIC memóriába, vagy hibajelzéssel visszatér. A parancsot is előbb átadja az interpreternek, s ha hibátlan, végrehajtja.	RETURN Lezárja a programsor szerkesztését és azt a Basic memóriába viszi. A parancsot azonnal átadja az interpreternek.
CAPS	Nagybetű/kisbetű váltása SHIFT-tel: mindig nagybetűre vált, ha az előző állapot kisbetű volt; CTRL-lal: grafikus jeleket ad (ezek a billentyűzeten nincsenek ugy feltüntetve, mint a C=64-nél).	C= és SHIFT együttes lenyomása váltja ki ugyanezt.
CTRL és 3 CTRL és 4 CTRL és 1	EOF (end of file); hangjelzés; a kurzor az első alfanumerikus billentyű lenyomására eltünik a képernyőről, majd a CTRL 1 lenyomására ismét láthatóvá válik. Tehát megállitja és inditja a képernyő változását.	Szinvezérlés.
ESC	A következő billentyű nemzetközi karakterként, vagy grafikus karakterként lesz megjelenitve, és akkor lesz végrehajtva, ha a képernyőn kinyomtatódik. Programozható billentyű pl: 10 PRINT"[ESC]" (A nyil előállitása: az ESC megnyomását követően, a SHIFT nyomvatartva, a CLEART megnyomjuk.) hatása: automatikus képernyő törlés.	Nincs.
CLR SET TAB	CTRL-lal vezérelve az adott pozicion törli a tabulálást; SHIFT-tel vezérelve beállitja a tabulálást önmagában pedig tabulál.	Nincs.
RESET	A programot és a változókat nem törli, de az utasitásszámlálót a program elejére állitja. Tehát a CONT paranccsal nem folytatható a program.	RUN/STOP és RESTORE alaphelyzetbe állitja a gépet ; a program és a változók nem vesznek el.
	Inverzre váltás.	RVS/ON és RVS/OFF.
OPTION SELECT START HELP	Programozható funkcióbillentyűk.	F1 - F8 billentyűk némileg hasonlóak.

3. táblázat

EGYSÉGAZONOSÍTÓK

A személyi számítógépeknél a be- és kivitelnél minden esetben tudni kell, hogy milyen perifériáról, illetve eszközről van szó. Éppen ezért mindig meg kell adni a hozzájuk tartozó alfanumerikus jelet. Az adott gépek esetében ezeket a *4. táblázat* mutatja.

NYOMTATÓ

Az ATARI 1029 80 karakteres, 5×7 pontos mátrixnyomtató. Alapállapotban az irásszélessége 10 karakter/inch 80 oszlopban, nyújtott irásmódban pedig 5 karakter/inch 40 oszlopban.

LINESPACE (a sor és az utána következő üres hely együtt): egy inchen 6 sor fér el az alap üzemmódban, 9 sor grafikus üzemmódban, ahol egy sort egyszerre nyomtat ki. Sebessége 50 karakter/mp (a reguláris szélesség mellett). Beépitett interfész teszi lehetővé valamennyi ATARI géphez való illesztését. Hálózati tápegysége szintén belső tartozék. Öntesztkapcsolóval és hibajelzővel van kiegészitve.

A kézi papirtovábbitás előre és hátra is működik, akár különálló A/4-es lapokra is képes nyomtatni. Karakterkészlete 132-féle alfanumerikus jelből áll.

A C64 nyomtatói szintén 80 karakteresek, de 6×7 pontos mátrixszal dolgoznak. Nyomtatási sebességük az ATARI-éhoz hasonló: 50 kar/mp.

A 803 típus ennél valamivel gyorsabb. Papirt továbbitani csak előre képes, visszafelé nem.

LEMEZMEGHAJTÓ

Az ATARI 1050 lemezegység ugyanolyan lemezekhez (mini

floppy: 5,25") való, mint a C64 VC—1541 meghajtó, de az előbbin szimpla és dupla sűrűségű írásmóddal egyaránt lehet dolgozni. Az *5. táblázat* az ATARI-nál és a Commodore-nál használatos lemezek felépítését, szervezését, illetve a meghajtó néhány jellemzőjét mutatja.

Az 1050-es meghajtó soros buszon, szinkroncsatornán keresztül csatlakozik az alapgéphez. Ismeri a DOS 2.0 OS-t, a DOS 2.5-öt és a DOS 3.0-át.

A DOS 2.0 OS csak szimpla sűrű lemezekkel dolgozik, ahol a lemezen 40 sáv, egy sávon 18 szektor, egy szektorban pedig 128 bájt található. A DOS 2.5-nél is 40 sávos a lemez, de itt egy sávon belül már 26 szektort találunk, egyenként 128 bájttal. A rendszer szimpla és dupla sűrűségű lemezeket egyaránt elfogad.

A szóban forgó két géptípus között az is különbségként mutatkozik, hogy az A—800 XL 1050 típusú meghajtónak hálózati táp-

	ATARI 800 XL		COMMODORE 64
Billentyűzet	K:		0
Nyomtató	P:		4, 5
Kazetta	C:		1
Képernyő	S:		3
Képernyő-szerkesztő	E:		-
Lemez meghajtó	D:	(D1:,D2:)	8,9,10,11,
Plotter	*		6
RS-232 csatorna	R:		2

* Nincs információ

4. táblázat

5. táblázat

6. táblázat

	ATARI 810, 1050		COMMODORE 1541
	szimpla sűrűség	dupla sűrűség	
Sáv/lemez	40	40	35
Szektor/sáv	18	13	17-21
Blokk/lemez	90	130	683
Szektor/lemez	720	1040	683
Szektor/blokk	8	8	1
Bájt/szektor	128	128	256
Bájt/blokk	1024	1024	256
Írási/olvasási sebesség	19200 Baud		3840 Baud
ROM	4 Kbájt, és az alapgép RAM-jából még 128 bájtot felhasznál.		16 Kbájt operációs rendszer + 2 Kbájt RAM
Mikroprocesszor	6507		6502

A lemezek teljes kapacitása (bájtban):

ATARI 800 XL				COMMODORE 64
DOS 2.0 S	DOS 2.5	DOS 3.0	1541 DOS	
90468	133120	*	174848	

* nincs információ

MIKROSOFT standard	A-800 XL	C=64
CALL gépi rutin hívása	USR(cim)	SYS cim
CHAIN [MERGE]..[ALL] programláncolás	RUN sorszám	--
CMD output átcimzése	--	CMD fájlszám
CLOSE fájl lezárása	CLOSE [#]	CLOSE fájlszám
DEF FN függvénydefinició	--	DEF FN
DIM tömbdimenzionálás	DIM, vagy COM	DIM
IF..THEN..[ELSE] feltételes elágazás	IF..THEN..[ELSE]	IF..THEN..
INKEY$ billentyűzetolvasás	GET#	GET
LEFT$ kiemelés balról	.$(tól,ig)	LEFT$(.$,karakt.szám)
LLIST lista printerre	LIST"P:"	OPEN 4,4:CMD4:LIST
LOAD programbetöltés	LOAD"egység: "	LOAD" "[,egységszám]
LOG logaritmus	LOG( ) 'e' alapú CLOG( ) 10 alapú	LOG( )
LPRINT nyomtatás printerre	LPRINT	PRINT#
MID$ kiemelés középről	.$(tól,ig)	MID$(.$,tól,karakter)
ON ERROR GOTO elágazás hiba esetén	TRAP	--
OPEN fájl megnyitása	OPEN#	OPEN fájlszám
RESTORE DATA mutató állitása	RESTORE [sorszám]	RESTORE
RIGHT$ kiemelés jobbról	.$(tól,ig)	RIGHT$(.$,karakter)
SAVE program mentése	SAVE"egység: "	SAVE" "[,egységszám]
SYSTEM op.rendszer hivása	BYE	--
USR gépi rutin hivása	USR(cim)	USR(paraméter)
WAIT várakozás feltételre	--	WAIT
WIDTH képernyőszélesség	POKE 82,bal margó POKE 83,jobb margó	--

Sajátos utasítások:		
Program behivása listafájlból	ENTER	--
Hangcsatorna beállitása	SOUND	--
Visszatérési informa. eltávolitása verem tetejéről	POP	--
Egy bájtot tesz egy változóba	PUT	--
A következő bájt címének meghatározása	NOTE	--
A következő bájt címére mutat	POINT	--
Grafikus mód választása	GRAPHICS	--
Szinárnyalat és fényi beállitása	SETCOLOR	--
Szinregiszter kiválasztása	COLOR	--
Egy pont v.karakter felhelyezése a képernyőre	PLOT	--
Képernyőpozició kiválasztása	POSITION	--
Karakter kiemelése képernyőpozicióról	LOCATE	--
Egyenes rajzolása kurzorpoziciótól	DRAWTO X,Y	--
Füzér kezdetének memória címét adja	ADR(.$)	--
Játékvezérlő utasitások	PADDLE, PTRIG, STICK, STRIG	--

7. táblázat

egysége kivül helyezkedik el (míg a C64-es 1541-es meghajtójánál és általában a használható többi meghajtónál is a hálózati tápegységet egy házba épitették a motorral és az elektronikával), igy az ATARI-gépeknél kevésbé kell az üzemi túlmelegedés veszélyével számolni, s ezért alkalmasabbak folyamatos (10—14 órát meghaladó) üzemre.

A BASIC INTERPRETER

A BASIC interpreter (értelmező) ROM-ban helyezkedik el, és a memória legfelső 16 kbájtjából 8 kbájtot foglal el. (Helyzetéről egy későbbi cikkben lesz szó.) Az interpreter a MICROSOFT BASIC értelmezővel összehasonlitva egy egyszerüsitett változat, amelyet kiegészitettek néhány speciális utasitással.

A *6. táblázat* az A—800 XL és a C64 BASIC eltérő kulcsszavait mutatja. Sajátos utasitások is szerepelnek az ATARI készletében *(7. táblázat)*.

Az egyes utasitások azonossága még nem jelenti azt, hogy a két gépen azonos koncepció szerint is működnek. Erre a kérdésre, vagyis az interpreter működési sajátosságaira is a folytatásban térünk vissza.

JÁNOSA—PAÁL—SÜTŐ

A Mezőgazdasági Épitő és Szerelőipari Szövetkezeti Vállalat

IBM PC XT/AT számitógépekre készült alább felsorolt

Software-csomagokat ajánlja megvételre:

1. Anyaggazdálkodási rendszer (anyag, fogyó, göngyöleg, félkész-, késztermék)
- minimum-maximum készletek
- elfekvő készletek
- árkülönbözet-számítás
- főkönyvi kimutatás
- leltárkészítés
- utókalkuláció

2. Munkaügyi nyilvántartás, statisztika
- létszámjelentések
- szabadság, pótszabadság
- tömegszervezeti tagsági díjak

3. Bérszámfejtés, bérstatisztika
- havidíjas, órabéres, részmunkaidős, nyugdíjas dolgozók számfejtése
- bérkifizetési lap nyomtatása
- címletkészítés (kifizető hely szerint)
- bérfelosztás
- statisztikai adatok göngyölése

4. Állóeszköz-nyilvántartás
- értékcsökkenés-számitás
- állóeszköz-állományok alakulása

5. Pénzforgalom
- vevő-szállító nyilvántartás
- pénztár-, bankbizonylatok feldolgozása
- számlák

6. Főkönyvkészítés

Az állományok naprakészen állandóan lekérdezhetők, módosithatók, listázhatók.
Az álrendszerek szervesen kapcsolódnak egymáshoz.

Vállalatunk winchesterek bérkimentését 20 Mbyte-os streamerrel folyamatosan vállalja.

7. C—64 számitógépekre készült költségvetési programok

8. Több típusú számitógép elhelyezésére alkalmas számitógépszekrény

MEZÉPSZER
2040 Budaors, Építők útja 2-4.

Mesterséges értelem II.

Alkalmazott intelligencia

AI = Artifical Intelligence, azaz mesterséges intelligencia. Ezek a betűk bő két évtizedig kisszámú lelkes tudós csendes laboratóriumokban végzett, a gyakorlattól távolinak látszó kutatásait jelentették. Napjainkban ezt a betűszót egyre gyakrabban másképpen, Applied Intelligence-nek, azaz alkalmazott intelligenciának fejtik. Jelzi ez is, hogy mostanában a kutatók egyre több gyakorlati alkalmazást ígérő módszert és konkrét eredményt tettek az emberiség asztalára.

Az első sakkprogramok is e kutatások keretében keletkeztek. (Lapunk Sakkprogramozás rovatában sok érdekességgel találkozhattak már olvasóink, és az e téren úttörő eseményekről továbbra is rendszeresen beszámolunk. — A szerk.) Az emberi beszédet értelmezni képes, a látás mint funkció által bizonyos problémák megoldására alkalmas programok mind-mind az ún. mesterséges intelligencia technikákat alkalmazzák. Ezek általában is a szoftvertermékek intelligenciájának tendenciózus növekedéséhez vezetnek, következésképpen a velük való feladatmegoldás jelentős megkönnyítéséhez, így és ezáltal sokkal szélesebb körű alkalmazhatóságukhoz. Újabb és újabb területekre hatol be a számítógép, mint a fejlődés tárgyi minimuma. Mindez már mutatja, hogy gyökeresen megváltozik az ember-gép kapcsolat. **A természetes nyelvet legalább részlegesen megértő programok** mérföldkövet jelentenek; egyik alkalmazási területük a más feladatokra készített programok felhasználói interfészeként való működés. Ez azt jelenti, hogy a különböző funkciókat ellátó programokat természetes emberi nyelven vezérelhetjük. Ez hasonlíthatatlan jelentőségű többek között a mindenki által elérhető adatbázisok lekérdezésében. Ilyen például egy lakáscsere-adatbázis.

Az oktatásban is jól hasznosítható az ilyen lehetőség. Egy adatbázis-szolgáltatást mutat be az *1. ábrán* látható képernyőállapot, amely a Borland cég PROLOG-ban készült, az angol nyelvet részben értelmezni képes adatbázis-kezelő programjának egy használat közbeni helyzetét tükrözi. Az adatbázis az Amerikai Egyesült Államok geográfiai adatait tartalmazza, mint ahogy ezt a *2. ábra* mutatja. (Sajnos a képernyődumpokon a grafikus karakterek helyett a printer mindenféle betűket nyomtat, úgy hogy a grafikus karaktereket kicseréltem I-re, = -re, *-ra stb. — A szerző.)

Visszatérve a bemutatott rendszerre, egy másik példával megvilágítjuk az ilyen és hasonló szolgáltatások nagymérvű rugal-

```
-------------GEOBASE: Natural language interface to U.S. geography-------------
I
IQuery: what is the population of the state with the highest point ?
I401800 citizens
I
IQuery: which is the highest point in california ?
Imount whitney
I
IQuery: which is the longest river in the state with the lowest point ?
Icolorado
I
IQuery: what is the population of the biggest city ?
I7071639 citizens
I
IQuery: give me which rivers run through states that border the state whose
Icapital is austin ?
Ineosho        washita       arkansas      st. francis   white
Imississippi   ouachita      pearl         sabine        red
Icanadian      cimarron      rio grande    san juan      gila
Ipecos
I16 Solutions
I
IQuery:
--------------------------------------------------------------------------------
```

1. ábra. A Geobase-program (Borland)

```
-------------GEOBASE: Natural language interface to U.S. geography-------------
IQuery: what is the population of the biggest city ?
I7071639 citizens
I
IQuery: how many natives live in New York ?
I>> Unknown word: native
I                                    ----------------Language-----------------I
I                                    I1  Schema for relations
I                                    I2  Schema for the entity network
I                                    I3  Names of entities
I                                    I4  Synonyms for entities
I                                    I5  Alternative names for associations
I                                    I6  Words to ignore
I                                    I7  Units for attributes
ISynonym         Entity              I8  Alternatives for relation operators
I*************** ***************     I9  Words stating minimums
Ipeople          population          I10 Words stating maximum
Icitizen         population          -----------------------------------------I
Iinhabitant      population
Iplace           point
Itown            city
I
I
IPress any key to continue
--------------------------------------------------------------------------------
```

3. ábra. A Geobase-nyelvezet bővítése

2. ábra. Borland Geobase képernyőfordítása

```
-------------GEOBASE: Natural language interface to U.S. geography-------------
I
IKérdés:Mennyi annak államnak a népessége, amelyikben az Egyesült Államok
I   legmagasabb pontja található ?
I401800 citizens
I
IKérdés:Mi Kalifornia állam legmagasabb pontja ?
Imount whitney
I
IKérdés:Melyik a legalacsonyabb pontot tartalmazó állam leghosszabb folyója ?
Icolorado
I
IKérdés:Mennyi a legnagyobb város népessége ?
I7071639 citizens
I
IKérdés:Melyik folyók folynak át azokon az államokon, amelyek határosak azzal
I   az állammal, amelynek fővárosa Austin ?
Ineosho        washita       arkansas      st. francis   white
Imississippi   ouachita      pearl         sabine        red
Icanadian      cimarron      rio grande    san juan      gila
Ipecos
I16 Megoldás
I
IKérdés:
--------------------------------------------------------------------------------
```

4. ábra. A nyelvezetbővítés képernyőfordítása

```
-------------GEOBASE: Natural language interface to U.S. geography-------------
IKérdés:Mekkora a legnagyobb város népessége ?
I7071639 citizens
I
IKérdés:Hány bennszülött él New Yorkban ?
I>> Ismeretlen szó: bennszülött
I                                    ----------------Nyelv--------------------I
I                                    I1  Viszonyok sémája
I                                    I2  Adathálózat sémája
I                                    I3  Adatok megnevezései
I                                    I4  Adatnevek szinonímái
I                                    I5  Asszociációk alternatív nevei
I                                    I6  Figyelmen kívül hagyandó szavak
I                                    I7  Mértékegységek
ISzinoníma       Adatnév             I8  Relációoperátorok alternatív nevei
I*************** ***************     I9  Minimumot jelentő szavak
Iember           népesség            I10 Maximumot jelentő szavak
Iállampolgár     népesség            -----------------------------------------I
Ilakos           népesség
Ihely            pont
Ivároska         város
I
I
IPress any key to continue
--------------------------------------------------------------------------------
```

masságát. Ha a szövegelemző szegmens nem érti meg a kérdést, erről tájékoztatja a kérdezőt (*3.* és *4.* ábra). Ilyenkor ő párbeszédesen módosíthatja a nyelvelemzést végző programszegmenst: új rokon értelmű szavakat adhat meg, vagy szükség esetén akár új mondatszerkezeteket is definiálhat. Az ábrákon a módosítási lehetőségeket mutató ablakot és az adatbázis mezőneveinek aktuális rokon értelmű szavait látjuk. A 4. ábrából kitűnik, hogy a népesség szó mellett az ember, állampolgár, lakos szavakat is bevittük.

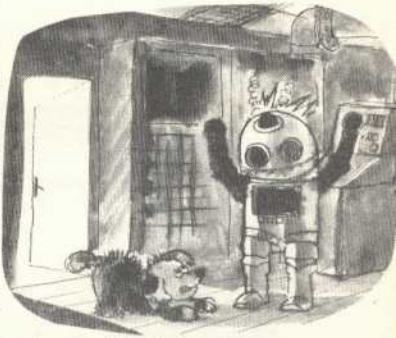
A beszédértelmezés technikája nemcsak a szorosan vett számítástechnikai, informatikai alkalmazásokban hasznosul, hanem bizonyos értelemben sokkal jelentősebb területeken is. Ilyen például — reménybelien — a nagy mennyiségű információ fordítása. Ez napjainkban, amikor a kereskedelmi és tudományos kapcsolatok egyre bővülnek, rendkívül nagy segítséget jelentene.

Japánban nagy erőket vetettek be olyan telefonrendszer kifejlesztésére, amelyen keresztül például kereskedelmi tárgyalásokat lehet folytatni automatikus fordítással oly módon, hogy az egyik fél Tokióban beszél japánul, a vonal másik végén a partner Párizsban franciául. A francia a japán fél válaszait is franciául kapja a fordítógéptől, és ugyanígy a japán beszélő is japán **válaszokat kap a fordítógép feléje eső oldaláról.**

Asimov híres sci-fi trilógiájában, az Alapítványban szerepel egy olyan írógép, amely a diktált szöveget leírja. Első pillanatra ez nem is tűnik igazán nehéznek, hiszen hangokat több-kevesebb megbízhatósággal felismerő rendszer már ma sem szenzáció. Viszont ezt a részeredményt a teljes feladat megoldásához hasonlítva — és az eszközöket ilyen céllal kipróbálva — rá kellett jönniük a bizakodóknak, hogy a szöveg nyelvtanilag helyes leírásához, az

esetleg félreértelmezett hangok korrigálásához (azaz a megbízható működéshez) legalább egy bizonyos szintig értelmezni is kell a szöveget. Márpedig **valós időben** ezt nem is olyan kis feladat végrehajtani. A megoldáson mindenesetre dolgozik a Kurzwell Applied Intelligence nevű cég, amely ezen az elven **működő írógép** gyártását tervezi.

A számítógépek beviteli rendszerén nemcsak a természetes beszéd értelmezése változtat. A vizuális információ gépi feldolgozása sok olyan területet nyitott meg a számítógépek előtt, ahol eddig megoldhatatlan nehézségek álltak az alkalmazás útjában. A gépi látás, alakfelismerés, tájékozódás legkézenfekvőbb felhasználási területe a robotika. Látványos sikert aratott például a Carnegie—Mellon Egyetem robotirányí-



5. ábra. A Helix Expert Systems cég Expert Edge nevű programjának demonstrációja

```
-------------------------------------------------------------------------------
Is cold   diagnosed                    IF  your nose   is   running            I
I                                                                             I
I                                      AND  you   are   sneezing              I
I                                                                             I
I                                      AND  you   have   a high temperature   I
I                                                                             I
I                                      AND  you   have                        I
I-----------------------------------------------------------------------------I
Idialogue window                       Iexpert edge   2nd Aug 00              I
I                                      Ifree  96%                             I
I                                      I-----------------------------------I
I                                      Imessage window                        I
I                                      I                                      I
I-----------------------------------------------------------------------------I
IWhat is the first object (if any):     a sore throat                         I
I                                                                             I
I                                                                             I
I                                                                             I
I                                                                             I
I                                                                             I
I=============================================================================I
ILearn:                                                                       I
I navigate,  add a rule, or  delete a rule,  rationalise knowledge            I
```

6. ábra. Az Expert Edge demonstrációs képernyőjének fordítása

```
-------------------------------------------------------------------------------
Imegfázás megállapítva                 HA   az orrodat törölgetni kell        I
I                                                                             I
I                                      éS   szipogsz                          I
I                                                                             I
I                                      éS   hőemelkedésed van                 I
I                                                                             I
I                                      éS                                     I
I-----------------------------------------------------------------------------I
Idialógus ablak                        Iexpert edge   2nd Aug 00              I
I                                      Ifree  96%                             I
I                                      I-----------------------------------I
I                                      Iüzenet ablak                          I
I                                      I                                      I
I-----------------------------------------------------------------------------I
IMi az első dolog (ha van egyáltalán):     rekedtség                          I
I                                                                             I
I                                                                             I
I                                                                             I
I                                                                             I
I                                                                             I
I=============================================================================I
ITanuló mód:                                                                  I
I navigáció , szabály bavitele , szabály törlése , ismeretek racionalizálása  I
```

```
I                Medical Adviser                                    I
I                                                                   I
IThe acceptance level is currently 70                               I
I                                                                   I
I                                                                   I
I                                                                   I
I                                                                   I
I-------------------------------------------------------------------I
IAre you sneezing a lot?                 Iexpert edge   2nd Aug 00  I
I100                                     Ifree  95%                 I
IDo you have a higher than normal        I----------------------------I
Itemperature?                            I                          I
I                                        I                          I
I-------------------------------------------------------------------I
IDo you have a higher than normal temperature?                      I
IMove along scale, press function key or select value to answer     I
I yes,                    maybe,                          no,       I
I100..............75..............50..............25..............0 I
I don't know  value  why  tell  but.                                I
I                                                                   I
I===================================================================I
IMain Menu:                                                         I
IPlease select one of the following commands:  Advise  Learn  Change  Disk.  I
```

7. ábra. Egy diagnosztikai rendszer alkalmazási példája

```
I                Orvosi Tanácsadó                                   I
I                                                                   I
IA beállított elfogadási szint 70  ( a rendszer u.i. a válaszokhoz valószí-  I
I                                  nűségi értékeket rendel hozzá, amelyeket  I
I                                  azután a döntésekben figyelembe vesz. Itt  I
I                                  ez 70% . - a szerző )            I
I                                                                   I
I-------------------------------------------------------------------I
ISokat szipogsz ?                        Iexpert edge   2nd Aug 00  I
I100                                     Iüres  95%                 I
IMagasabb a hőmérsékleted a              I----------------------------I
Inormálisnál?                            I                          I
I                                        I                          I
I-------------------------------------------------------------------I
IMagasabb a hőmérsékleted a normálisnál ?                           I
IÁllítsd be a skálán , használd a funkcióbillentyűt , vagy adj meg értéket  I
I igen,                   talán,                          nem,      I
I100..............75..............50..............25..............0 I
I nem tudom    érték    miért    mondd    de                       I
I                                                                   I
I===================================================================I
IFő Menü:                                                           I
IVálassz egyet a következő parancsok közül :  Tanács Tanulás Módosítás Diszk  I
```

8. ábra. Az orvosi tanácsadás lefordított képernyője

tó rendszere, amely ha nagyon lassan is, de képes a rábízott robotot egy ismeretlen, akadályokat tartalmazó területen önállóan, az akadályok kikerülésével átvezetni. Van egy eset azonban, amikor a rendszer csődöt mond, és **a robot megbénul.** Ez akkor következik be, ha a laboratórium kutyája betéved a kísérleti területre. A VAX 11/780-as számítógépen működő rendszer ugyanis jelenleg képtelen egy ilyen gyorsan mozgó, változó objektumot felismerni, elhelyezni, azonosítani, következésképpen a mellette való elhaladás stratégiáját sem képes kidolgozni.

A mesterséges intelligencia kutatások legjobban reklámozott eredménye a szakértői rendszerek kifejlesztése. A szakértői rendszerek lényegében olyan programok, amelyek az embertől a problémamegoldási feladatok egy részét átveszik, és több-kevesebb segítséggel elvégzik. Miért nevezhetők e programok mégis szakértői rendszernek, ha látszólag ilyen „önállótlanok"? Részletesebb összehasonlításra van szükség a kérdés megválaszolásához. A hagyományos számítógép-programok az embert a problémák megoldásában csak támogatták adatok szolgáltatásával, illetve számítások elvégzésével. Ilyenek voltak még a legfejlettebb alkalmazói/felhasználói szoftverek is, úgymint az adatbázis-kezelő programok — ezek adatok gyors visszakeresését és feldolgozását segítik — és a táblázatkezelők (ún. spreadsheet programok) is —, melyek erőforrások optimális elosztásának próbálkozásokkal történő meghatározását támogatják. Viszont a programoknak ez az új generációja automatikusan végigvizsgálja a — természetesen előzőleg beprogramozott — választási lehetőségeket adott szempontok szerint, és elvégez rutinszerű problémamegoldási feladatokat. Ezek azon alapszanak, hogy az intelligens viselkedés rendkívül sok ismeret alkalmazását igényli. Olyan mennyiségű ismeretet jelenleg nem tudunk számítógéppel kezelni, amelyre egy minden szituációban működőképes rendszernek szüksége lenne, de valamely szűk terület információmennyiségével a számítógép többnyire sikerrel megbirkózik. A program a rendelkezésére álló információból következtetéseket tud levonni, sőt az ismeretek szokatlan szempontból való megvizsgálásával még készítőit is meglepheti. Természetesen ehhez a program ismeretbázisának pontosnak kell lennie.

A szakértői rendszer elnevezés tulajdonképpen nem helyes, hiszen az alkalmazókat megtéveszthetí. A tudásalapú rendszer (knowledge-based system) vagy a szabályalapú rendszer (rule-based system) igazabb elnevezés, hiszen ezek a rendszerek a készítőik által a terület szakértőivel együttműködve lefektetett és a program ismeretbázisába bevitt elveket és szabályokat alkalmazzák problémamegoldásra. Ha az elkészítés során valami tévesen került be, vagy valami kimaradt, később ez természetesen könnyen módosítható vagy pótolható, de ha egy probléma bizonyos vonatkozásával kapcsolatban a rendszer nem rendelkezik információval, akkor az emberi szakértővel szemben teljesen téves döntést fog hozni. A szakértői rendszerek fő alkalmazási lehetőségei a következők.

— **Hibakeresés — tanácsadás.** A műszaki berendezések hibáinak felderítésétől és a javítási tanácsok adásától egészen az emberi szervezet bajainak kereséséig, azaz az orvosi diagnosztikáig és a kezelési, gyógyítási módok előírásáig terjedhet. Rendkívül sikeres például a Stanford Egyetemen kifejlesztett MYCIN rendszer, amely fertőzések azonosításában és a gyógyszerrendelésben nyújt segítséget az orvosnak.

Az *5.* és *6. ábra* az előbbihez hasonló szakértői rendszert mutat be készítése közben. Az ábrán egy orvosi rendszer által tartalmazott szabály definiálását láthatjuk. A szabály azt állítja, hogy ha valakinek sokat kell orrot fújnia, szipognia, láza van és fáj a torka, akkor egy megfázás nevű betegséget állapíthatunk meg nála. A *7.* és *8. ábra* az előbb definiált szabály alkalmazását mutatja.

Az aktuális kérdésre, vagyis hogy magasabb-e a hőmérséklete a normálisnál, a gép előtt ülő személy válaszolhat igennel, talánnal, nemmel vagy beállíthat egy értéket a skálán. Ha ezek egyike sem volna helyes, akkor a skála alatti menüre térhet, ahol újabb lehetőségeket kap a válaszadásra, illetve ő is kérdezhet a rendszertől. Válaszolhatja azt, hogy nem tudom, megadhatja a konkrét hőmérsékletét, megkérdezheti, hogy miért érdekli a rendszert a hőemelkedés mértéke, továbbá felszólíthatja a rendszert újabb információk adására (például: mennyi a normális hőmérséklet?), és még ki is egészítheti választ (a "de" való az utóbbira).

— **Tervkészítés.** A szakértői rendszerek bonyolult, sok kölcsönhatással bíró rendszerek tervezésében, viselkedésük előrejelzésében is teret nyertek — például a **pénzügyi,** gazdasági, tervezési feladatokban vagy a műszaki tervezésben. Különösen erős az aktivitás a nagy bonyolultságú (több százezer tranzisztort tartalmazó) integrált áramkörök tervezését és **tesztelését végző** szakértői rendszerek terén. Ezek között van már olyan is, amely számítógépek központi egységeit és mikroprocesszorokat teljes egészében képes megtervezni (Design Automation Assistant).

— **Adatelemzés.** Ezek a rendszerek hatalmas mennyiségű adatból a lényeges információk kivonását, illetve az adatmennyiség bizonyos szempontból való elemzését végzik. Gondoljunk például légi felvételek elemzésére az ásványkincsek előfordulási valószínűsége szempontjából, vagy esetleg **próbafúrások mintáinak elemzésére** (PROSPECTOR).

A szakértői rendszerek alkalmazásának fő előnye a vezérlés és a felhasznált ismeretbázis fokozottabb elkülönüléséből adódóan a hagyományos rendszereknél nagyobb rugalmasság. Az információbázis maga ugyanis igen könnyen bővíthető, illetve a körülmények megváltozása esetén módosítható. A rendszer bővíthető tudásbázisa ideális az *inkrementális programfejlesztéshez.* A szakértői rendszerek lehetővé teszik bizonytalan vagy nem teljes információ kezelését is.

Van a mesterségesintelligencia-kutatásoknak más irányú eredménye is. A gondolkodási folyamatok elemzésével körvonalazódik, hogy vajon melyek lehetnek a gondolkodás alapvető jellegzetességei; ezek minden értelmes lény számára valószínűleg közösek. Ily módon az esetleges találkozáskor földön kívüli intelligenciával ez lehetne a kiindulás a kommunikáció megteremtéséhez. Ezzel foglalkoztak például Marvin Minsky és társai a SETI (Search for Extraterrestrial Intelligence = a földön kívüli intelligencia keresésének programja) keretében.

A számítástechnika története során egyre magasabb szintű nyelvek születtek és terjedtek el. Az egyre magasabb szintű nyelv használata tulajdonképpen azt jelenti, hogy a program feladatát, specifikációját a géptől egyre távolibb, az emberhez egyre közelibb módon fogalmazzuk meg: a nyelvek hierarchiájában feljebb haladva egyre kevesebb információra van szükségünk a gép konkrét működéséről.

A gépiintelligencia-kutatások hozták a felszínre a logikai programozás elvét, amelyet a PROLOG nyelv valósít meg. A PROLOG-ban a programozónak már nem kell a vezérlési struktúrát explicit módon definiálnia, hanem logikai szabályok formájában írja le a törvényszerűségeket, amelyek szerint a programnak működnie kell. Ezért ezt leíró programozásnak is nevezik. Az automatikus programozás végül is a programspecifikáció egyre magasabb szinten, egyre általánosabb módon való megadását jelenti, amelyből a mind bonyolultabb és intelligensebb *programozási környezetek,* interpreterek és szerkesztőprogramok — ezek szintén felhasználhatják a mesterségesintelligencia-kutatások eredményeit — előállítják, majd futtatják az aktuális gépi szintű programot.

Szintén ezeket az eredményeket alkalmazzák az intelligens felhasználói interfészek készítésében. Ezek olyan programok, amelyek figyelik a felhasználó viselkedését a programmal való munkája közben, és fokozatosan a számára kényelmesebb módon működnek. Például a kérdésre adott válaszokból és azok gyorsaságából **következtetnek a felhasználó gyakorlottságára.** A kezdők számára több segítséget nyújtanak: bőbeszédűbb lesz a dialógus; a gyakorlott felhasználótól viszont például már rövidítéseket is elfogadnak. Ezáltal a más-más képzettségűek és képességűek egyaránt a szemszögükből nézve fokozott hatékonysággal tudják a rendszert használni.

Az interfész intelligenciája természetesen nem merül ki ebben. A különböző felhasználók munkáját bizonyos ideig megfigyelve, ismétlődő tevékenységeiket személy szerint (a felhasználói azonosító szerint) felismeri, s bizonyos idő alatt megtanulja, hogy automatikusan rátérhet a következő szokásos lépésre. Általánosságban is figyeli a legvalószínűbb tennivalót: például egy változó szerepeltetése után figyelmeztet a deklaráció szükségességére, illetve el is végzi. Természetes, hogy ilyesmi által tovább javul az ember-gép kapcsolat.

Ilyen alkotórészeket is tartalmazó komplex rendszer például a katonai fejlesztésű AMAS. Ez egy pilótákat támogató rendszer, amely a cirkálórakéta vezérlőrendszeréhez hasonló feladatokat lát el: a bevetés során önállóan értékeli a gépet fenyegető veszélyeket, és akár a pilóta eszméletlen állapotában is képes biztonságos pályán elvezetni a gépet. Ebbe beletartozik a légelhárító fegyverek felismerése, az elfogó vadászgépek és rakéták pályájának folyamatos figyelése, továbbá az ezeknek megfelelő manőverek végrehajtása. Folyik az ismeretlen terepen önállóan tájékozódni és haladni képes járművek fejlesztése is. Az egyik ilyen rendszerrel történt meg, hogy a kísérleti út mellett álló egyik fa árnyékát útkanyarnak értelmezte, és a megfelelő ponton tökéletes fordulót végrehajtva **megpróbált felmászni** a fára. Ebből is látszik, hogy a fejlesztések még csak kezdetleges stádiumban vannak, de előbb-utóbb a polgári világban is természetes lesz a villamost vagy akár személyautót emberi közreműködés nélkül vezető számítógép alkalmazása.

A mesterségesintelligencia-technikák, együttesen a mikroelektronika további fejlődésével, várhatóan a számítástechnikai megoldások olyan mértékű szélesedését okozzák, hogy ez az élet minden területén észrevehető lesz. A gépek több és több kellemetlen feladat végzésétől szabadítják meg az embert, és az alkotó munkát az eddiginél sokkal jobban támogatják majd.

SOMOGYVÁRI KÁROLY

FOGALMAK

szakértői rendszerek: tanácsadásra, esetleg önálló problémamegoldásra képes, kódolt emberi ismereteket felhasználó programok

spreadsheet: táblázatok kezelésére készített programok, amelyek nyilvántartják a táblázat különböző rovatai közötti összefüggéseket, és szükség szerint újraszámítanak rovatokat

inkrementális programfejlesztés: a programkészítés olyan módszere, ami lehetővé teszi a program képességeinek folyamatos növelését a program szerkezetének lényeges módosítása nélkül. Erre az ad módot, hogy a programot egy váz köré, ismeretek későbbi bevitele szempontjából nyitottan építik fel

ismeretalapú, tudásalapú rendszer: explicite — például szabályokban — kódolt ismeretekre támaszkodó rendszer (lásd szakértői rendszer)

automatikus programozás: programok generálása be- és kimeneti példasorokból, működési példákból, illetve igen magas szinten akár természetes nyelven megadott specifikációból is

Programozzunk PASCAL-ban: scroll

ZX–SPECTRUM

Előre kell bocsátanom: elfogult vagyok a Sinclair ZX-Spectrum számítógéppel szemben még akkor is, ha tudomásul kell vennem, eljárt felette kissé az idő. Azt hiszem, mégis egy dologban a kategóriájában egy darabig még versenytárs nélkül marad: ez pedig az a tény, hogy olyan kiváló fordítóprogramok készültek Spectrumra, amelyek egyedülálló eszközzé teszik azok számára, akik a gépet számítástechnikai önképzésre akarják használni, vagyis a jó értelemben vett amatőrök számára.

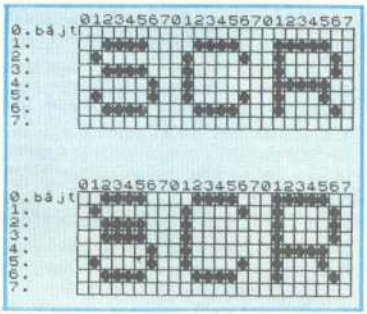
A Spectrumra kitűnő PASCAL, C, micro-Prolog, Lips, FORTH fordítóprogram kapható, és tapasztalatom szerint ezek a programok közkézen is forognak hazánkban. Mégis kevesen használják ezeket az eszközöket, aminek valószínűleg az a legfőbb oka, hogy nincs irodalom, melynek alapján a kezdők elindulhatnak, a haladóbbak pedig további ösztönzést kaphatnának.

Ezen a helyzeten szeretnék változtatni valamicskét a magam szerény eszközeivel. Remélem kísérletem nem marad magányos. Az alábbiakat elsősorban olyan olvasóknak szántam, akik a BASIC programozással tisztában vannak, valamennyire ismerik a Spectrum memóriatérképét, és kedvük támadt kipróbálni a PASCAL nyújtotta előnyöket.

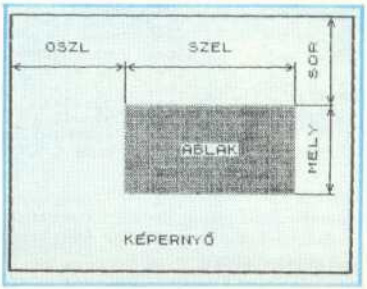
A programot Hisoft PASCAL-ban írtam. A Hisoft cég három ismert PASCAL-ja a HP4S, a HP4T és a HP4T16M ugyanannak a fordítóprogramnak egyre fejlettebb változata, bármelyik alkalmas a példaprogram kipróbálására.

1. lista

```
 10 REM SCROLL DEMO BASIC-SEN
100 CLS : PRINT "SCROLL DEMO"
110 FOR k=1  TO 8
120 LET a=16384
130 FOR j=0 TO 6
140 LET b=a+j*256
150 FOR j=b TO b+11
160 IF j<>7 THEN  POKE j,PEEK (j+256)
170 IF j=7 THEN  POKE j,0
180 NEXT j
190 NEXT j
200 NEXT k
```



1. ábra



2. ábra

Egy ilyen rövid cikk keretein belül nincs hely arra, hogy a PASCAL nyelv felépítését, szintaktikai szabályait is ismertessük. Akit ez érdekel, szerezze be K. Jensen és N. Wirth: A PASCAL programozási nyelv, illetve Varga Imre: PASCAL Spectrumra és Commodore 64-re c. könyvét. Az elsőből a PASCAL nyelv tanulható meg, az utóbbi pedig tartalmazza a Hisoft PASCAL-változatok részletes kezelését.

Most egy olyan feladat megoldását szeretném bemutatni, amelyhez a BASIC az interpreter lassúsága miatt eleve alkalmatlan. Ez a feladat olyan eljárások készítése, melyekkel a képernyőn egy kijelölt ablak tartalmát le, fel, jobbra és balra lehet görgetni, azaz „scrollozni", de nem karakterenként, hanem képelemenként (pontsoronként).

Lássuk először magát a problémát! Írjunk olyan BASIC programot, amely a képernyő bal felső sarkába írt feliratot felfelé pontsoronként kigörgeti. A program az *1. listán* látható. Hogyan működik?

Tudjuk, hogy a Spectrum minden karaktere 8*8 pontból épül fel. Ennek a memória RAM területén 8 bájt felel meg, melyek mindegyike a karakter egy-egy vízszintes pontsorát tartalmazza. A fekete-fehér képernyő tartalmát leíró memóriaterület a 16384 címen kezdődik, és 8*32*24=6144 bájtból áll. (8=egy karakter bájtjainak száma; 32=egy sorban ennyi karakter fér; 24=a sorok száma, beleértve az alsó, szerkesztő területet is. Az attributumokkal, vagyis a színekkel stb. most nem foglalkozunk.)

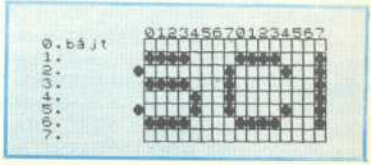
A 16384 címen a 0. sor 0. karakterének legfelső pontsorát leíró bájt van. Ennek a karakternek a következő pontsora 256-tal nagyobb című memóriarekeszben van; a 16385. rekesz a sorban következő karakter felső pontsorát kódolja. Ha most a felirat egy pontsornyi felfelé léptetését akarjuk megoldani, nézzük meg az *1. ábrát.* Itt a felirat raszteres felnagyított képének részletét látjuk úgy, ahogy a képernyőn megjelenik, és nem úgy, ahogy a memóriában tárolva van. Felfelé görgetéskor az első pontsort (bájtot) be kell töltenünk a 0. bájtba, majd a másodikat a 1-be, a harmadikat a 2-be, és így tovább az utolsó sorig.

Ekkor tevékenységünk látszólag véget ér, mert a nyert kép valóban az első karaktersor görgetett alakja. Azonban ez csak azért látszik így, mert karaktereink alsó pontsora üres. Ha itt is volna fekete pont valahol, az nem mozdulna el. Ahhoz, hogy helyesen haladjon a görgetés, az utolsó pontsort ki kell törölni.

Az 1. ábra alsó részén a görgetés egy közbenső fázisa látható: a 0—2. sort a program már felfelé léptette, a 3—7. sor még az eredeti helyén található.

Mintaprogramunkban a legbelső, J ciklus a felirat karaktereinek száma szerint fut, és egy feliratnyi szélességű pontsort léptet. A középső, I ciklus megismétli ezt az egykarakternyi magasságú felirat mind a hét pontsorára. Az I ciklus egyszeri lefutása a feliratot egy pontsorral felemeli. A leg-

3. ábra



```
 10 {$L-}
 20 PROGRAM SCROLLDEMO;
 30 VAR I,J,K,II,JJ,JJ,JI:INTEGER;
 40
 50 PROCEDURE UPSCROLL(SOR,OSZL,MELY,SZEL:INTEGER;
 51 RELUXA:BOOLEAN);
 60 VAR I,J,K,L,M,N:INTEGER;
 70     S,Z:ARRAY[0..31]OF CHAR;
 80 BEGIN
 90   FOR I:=0 TO 31 DO S[I]:=CHR(0);
100   FOR I:=SOR+MELY DOWNTO SOR+1 DO
110   BEGIN
120     J:=16384+OSZL+32*(I MOD 8)+(I DIV 8)*2048;
130     FOR K:=0 TO 7 DO
140     BEGIN
150       M:=J+K*256;
160       N:=0;
170       FOR L:=M TO SZEL+M-1 DO
180       BEGIN
190         IF K=0 THEN Z[N]:=PEEK(L,CHAR)
200         ELSE POKE(L-256,PEEK(L,CHAR));
210         IF K=7 THEN POKE(L,S[N]);
220         N:=N+1
230       END
240     END;
250     IF RELUXA=FALSE THEN S:=Z
260   END
270 END;{UPSCROLL}
280
290 PROCEDURE DOWNSCROLL(SOR,OSZL,MELY,SZEL:INTEGER;
291 RELUXA:BOOLEAN);
300 VAR I,J,K,L,M,N:INTEGER;
310     S,Z:ARRAY[0..31]OF CHAR;
320 BEGIN
330   FOR I:=0 TO 31 DO S[I]:=CHR(0);
340   FOR I:=SOR TO SOR+MELY DO
350   BEGIN
360     J:=16384+32*(I MOD 8)+OSZL+(I DIV 8)*2048;
370     FOR K:=7 DOWNTO 0 DO
380     BEGIN
390       M:=J+K*256;
400       N:=0;
410       FOR L:=M TO SZEL+M-1 DO
420       BEGIN
430         IF K=7 THEN Z[N]:=PEEK(L,CHAR);
440         IF K>0 THEN POKE(L,PEEK(L-256,CHAR));
450         IF K=0 THEN POKE(L,S[N]);
460         N:=N+1
470       END
480     END;
490     IF RELUXA=FALSE THEN S:=Z
500   END
510 END;{DOWNSCROLL}
520
530 PROCEDURE LEFTSCROLL(SOR,OSZL,MELY,SZEL:INTEGER;
531 RELUXA:BOOLEAN);
540 VAR I,J,K,L,M,S,SP,Z:INTEGER;
550 BEGIN
560   SP:=ADDR(S);
570   FOR I:=SOR TO SOR+MELY-1 DO
580   BEGIN
590     J:=16384+32*(I MOD 8)+OSZL+(I DIV 8)*2048;
600     FOR L:=0 TO 7 DO
610     BEGIN
620     M:=J+256*L;
630       Z:=0;
640       FOR K:=SZEL-1 DOWNTO 0 DO
650       BEGIN
660         S:=0;
670         POKE(SP,PEEK(M+K,CHAR));
680         S:=S*2+Z;
690         IF (S)255) AND (RELUXA=FALSE) THEN Z:=1
691         ELSE Z:=0;
700         POKE(M+K,PEEK(SP,CHAR))
710       END
720     END
730   END
740 END;{LEFTSCROLL}
750
760 PROCEDURE RIGHTSCROLL(SOR,OSZL,MELY,SZEL:INTEGER;
761 RELUXA:BOOLEAN);
770 VAR I,J,K,L,M,N,S,SP,Z:INTEGER;
780 BEGIN
790   SP:=ADDR(S);
800   FOR I:=SOR TO SOR+MELY-1 DO
810   BEGIN
820     J:=16384+32*(I MOD 8)+OSZL+(I DIV 8)*2048;
830     FOR L:=0 TO 7 DO
840     BEGIN
850       M:=J+256*L;
860       Z:=0;
870       FOR K:=0 TO SZEL-1 DO
880       BEGIN
890         S:=0;
900         POKE(SP,PEEK(M+K,CHAR));
910         N:=S;
920         S:=S DIV 2;
930         IF N=S+S THEN N:=0 ELSE N:=64;
940         S:=S+Z;
950         POKE(M+K,PEEK(SP,CHAR));
960         IF RELUXA=FALSE THEN Z:=N
970       END
980     END
990   END
1000 END;{RIGHTSCROLL}
1010
1020 BEGIN
1030   FOR I:=0 TO 21 DO
1040   FOR J:=0 TO 31 DO
1050   WRITE(CHR(134));
1060   II:=17;
1070   IJ:=1;
1080   JJ:=28;
1090   JI:=1;
1100   FOR K:=1 TO 4 DO
1110   BEGIN
1120     FOR J:=1 TO 3 DO
1130     FOR I:=1 TO 8 DO
1140     UPSCROLL(II,1,3,3,FALSE);
1150     FOR J:=1 TO 3 DO
1160     FOR I:=1 TO 8 DO
1170     LEFTSCROLL(IJ,28,3,3,FALSE);
1180     FOR J:=1 TO 3 DO
1190     FOR I:=1 TO 8 DO
1200     DOWNSCROLL(1,JI,3,3,FALSE);
1210     FOR J:=1 TO 3 DO
1220     FOR I:=1 TO 8 DO
1230     RIGHTSCROLL(18,JJ,3,3,FALSE);
1240     II:=II-4;
1250     IJ:=IJ+4;
1260     JI:=JI+4;
1270     JJ:=JJ-4
1280   END;
1290   WHILE INCH()' ' DO
1300 END.
```

2. lista

külső, K ciklus az I ciklus ismétlésével felfelé kigörgeti a képből az egész feliratot. Tessék kipróbálni!

Nos, a program működik, de borzasztó lassan. Minden illúzionk odavan, egyszóval BASIC-ben a feladat gyakorlatilag nem oldható meg.

Más a helyzet azonban a PASCAL esetében. A *2. lista* SCROLLDEMO nevű mintaprogramja négy eljárást (PROCEDURE-t) tartalmaz, melyek különböző irányú görgetésre szolgálnak. Az első, UPSCROLL nevű eljárás felfelé, a DOWNSCROLL lefelé, a LEFTSCROLL balra, a RIGHTSCROLL jobbra görget.

Mindegyik eljárás feje azonos paramétereket tartalmaz, melyek jelentése az alábbi (lásd a *2. ábrát* is):

— SOR, OSZL: az ablak bal felső sarkának karakterhelyzete,

— MELY: az ablak sorainak száma,

— SZEL: az ablak szélessége karakterekben számítva,

— RELUXA: logikai változó, mely a görgetés módját jelzi. Egy több sorból álló felirat ugyanis úgy görgethető, ahogy a közismert Reluxa redőny: vagy felhúzzuk az egész redőnyt, vagy egyszerre elfordítjuk a redőny összes lécét. A RELUXA változó TRUE vagy FALSE értékével a két lehetőség közül választhatunk.

Az UPSCROLL eljáráshoz magyarázatként a következőket fűzzük.

— A 90-es sor előállít egy képernyőnyi szélességű üres pontsort, melyet később a felirat feleslegessé váló alsó pontsorának törlésére fogunk használni.

— Az I ciklus az ablak sorainak számaszor ismétel.

— Az 12Ø-as sor kiszámítja az I. sor első karakterének bal felső pontsora bájtjának címét.

— A K ciklus egy karaktersor egy-egy pontsorát emeli.

— Az L ciklus az ablak oszlopainak számával ismétel.

— A BASIC példához képest eltérés, hogy RELUXA=FALSE esetén a felfelé kigörgetett Ø. pontsort, melyet az L ciklus átmenetileg a Z karaktertömbben tárol, betölti a következő karaktersor alsó, törölt pontsorába.

A DOWNSCROLL eljárás nem tartalmaz említésre érdemes újdonságot, működése az UPSCROLL alapján érthető. Mivel a K és I ciklus iránya megfordul, a görgetés ellentétesen halad.

Érdekes viszont a LEFTSCROLL működése. Itt az eddig jó, egyszerű stratégia csődöt mond, ti. az, hogy a görgetéshez elegendő az ablak mintáját leíró bájtokat megfelelően átrendezni. A *3. ábrán* látható, hogy a görgetés közben a bájton belül az egyes bitek (pontok) helye is megváltozik, a mintázat a bájton belül balra tolódik. Ehhez az assembly nyelvből ismert shift műveletek valamelyikére lenne szükség, ám ilyen nincs a PASCAL-ban.

Valamit mégis tehetünk. Kihasználhatjuk ugyanis a bináris (kettes számrendszerben működő) szorzó algoritmust a céljainkra. Az 1. ábra „S" betűjének 5. bájtját például azonkívül, hogy az ábrán látható mintázatot testesíti meg, értelmezhetjük kettes számrendszerbeli számként is, amikor a fehér pontok Ø-ás számjegynek, a fekete pontok pedig 1-es számjegynek felelnek meg. Ha ezt a számot, melynek tizes számrendszerbeli jelentését nem is fontos tudnunk, megszorozzuk 2-vel — mely bináris alakja 10 —, akkor az eredményt úgy számíthatjuk ki papíron, mintha decimális számokkal számolnánk:

$$\frac{0100010*10}{01000100=1000100}$$

Világos, hogy a számnak megfelelő mintázat (a 0-k és 1-esek egymáshoz képesti helyzete) nem változott, csak egy helyiértékkel feljebb tolódott el, vagyis balra. A 10-zel való szorzás miatt a szorzat 9 bites lett, a bal oldali, legnagyobb helyiértékű bit már nem fér a 8 bites bájtba, túlcsordul. Akik értenek a gépi kódú programozáshoz, tudják, hogy ilyen esetben a 9. bitet a mikroprocesszor F regiszterének egy bitje, az ún. carry flag tartalmazza.

Ha a képet alkotó karaktereket rendre egy-egy kétbájtos INTEGER szám kis helyiértékű bájtjába POKE-oljuk, a nagy helyiértékű bájtját pedig kinullázzuk, majd megszorozzuk az egészet kettővel, az eredmény kis helyiértékű bájtja az egy bittel balra tolt karakterbájt, a nagy helyiértékű bájtjának értéke pedig 0 vagy 1 lesz attól függően, hogy a mintázatból kiszorult-e egy értékes bit (a képen fekete pont) balra vagy nem.

A nagy helyiértékű bájt vizsgálata egyenértékű annak ellenőrzésével, hogy az egész INTEGER szám nagyobb-e 255-nél van sem. Mivel a PASCAL-ban van lehetőség kétbájtos INTEGER számok szorzására, eredeti feladatunk is megoldható.

A LEFTSCROLL eljárás a fenti algoritmust valósítja meg. Az S változó tartalmazza a szóban forgó INTEGER számot. Ennek a RAM tárbeli címét a Hisoft PASCAL ADDR függvényével határozhatjuk meg, és értékét az SP változóban tároljuk. A 670-es sorban az S változóba POKE-oljuk a soron következő karakterbájtot, majd a következő sorban megszorozzuk 2-vel, és hozzáadjuk az előző karakter balra lecsorgó bitjét, melyet Z-ben tároltunk. Ha a 690-es sor azt észleli, hogy a balra tolásnál lecsordult egy bit, Z értéke 1, egyébként 0 lesz. A 700-as sor az egy ponttal balra tolt bájtot visszatölti a képernyő-tárterületre, és így az elvégzett művelet eredménye láthatóvá is válik.

A RIGTHSCROLL hasonlóan működik, csak itt 2-vel való szorzás helyett 2-vel való egész típusú osztást kell végeznünk.

Az egyes PROCEDURE-ket úgy terveztük, hogy globális változókat ne tartalmazzanak. Ezért változtatás nélkül használhatók más programban is. Ebből a célból az 50—1010-es sorokat célszerű külön kimenteni: kazettás megnetofonnál a W1,1010,SCROLL paranccsal; az Interface 1-gyel és Microdrive-val rendelkezők pedig a P1,1010,1:SCROLL paranccsal. Az ilyen módon kimentett eljárások az include szolgálatás segítségével közvetlenül befordíthatók más programokba.

Az eljárások nem adnak védelmet hibás hívás esetén, például, ha a kijelölt ablak a képernyő területén részben vagy teljesen kívül esik. Ezt szándékosan tettük: a hibaellenőrzés feleslegesen hosszúvá és lassúvá tette volna a programot.

A mintaprogram az eljárások alkalmazását is bemutatja. Javasoljuk az olvasónak, hogy az ablakok változtatásával próbálja ki, hogyan viselkednek a PROCEDURE-k. Ha az ablak mérete túlságosan nagy, a görgetés sima mozgása szaggatottá válik.

Remélem, a bemutatott egyszerű példa gondolatébresztő, és ennek alapján az olvasó nehezebb feladatok megoldására is vállalkozni fog.

DR. KABOLDY PÉTER

ADOK-VESZEK-CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közületeknek gépelt soronként (40 karakter) 100,- Ft, magánszemélyeknek az első sor 50,- Ft, minden további sor 20,-Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

ADOK

Apple II.C számítógép (128 kbájt, floppy) eladó. Dr. Merchenthaler István, Pécs, Asztalos T. u. 2/22. Tel.: 14-334.

ATARI 800XL számítógép DATA settel áron alul eladó. Játékprogram is van. Érdeklődni lehet: Debrecen /52/ 13-292 telefonon.

Commodore 16 számítógép (1 éves, jó állapotban lévő) sok játékprogrammal eladó 14 000 Ft-ért. Érdeklődni lehet: Budapest, Nagyszentmiklós u. 26. 1171. /hét végén/

C 116-os számítógép magnetofonnal, német és magyar nyelvű dokumentációval, oktatókazettával 11 000 Ft-ért eladó. Cím: Wahl Csaba, Keszthely, Festetich u. 5. Ø360.

Commodore 1541 floppy eladó. **C64-es** programokat kazettán cserélnék. Micsik Gábor, Szeged, Szent Miklós u. 3. 6723.

SCISYS Superstar (28 k) típusú 2a fokozatú sakkszámítógép, tápegységgel együtt eladó. Nagy István, Sárvár, Entrzlüder út 29/c III. 11. 9600.

ZX81-es géphez (64 kb) bővítő olcsón eladó. Levélcím: Wágner Gyula, Budapest, Hidegkúti út 80/a. 1028.

Programok C Plus/4 bővített C16 számítógépekre FORTH nyelven (nem tinyforth!/ 450 szó, 5 screen, diszk, printer és kiemelt magnókezelő utasítások. C Plus/4 -> C64 összehasonlító futtató- és rendszerváltozó táblázat. Turbo programok /C64 turbo-tape kompatibilis változat is!/ turbo fájl kazettán fájlkezelést tízszeresre gyorsító program, BASIC bővítéssel. SYSTEM C Plus/4 II kbájt rendszerprogram BASIC és assembler fejlesztéshez. HEADER JUSTAGE: fejbeállító program. Disc-monitor. Turbocopy diszkhez és magnóhoz. C64-es program: MONITOR'PLUS /a háttér RAM-okat is kezeli/. A programok ára: 180-600 Ft, kazettán, utánvéttel, részletes kezelési utasítással. Kérésre ismertetőt küldök. Cím: Pelsőczi Gyula, Szilasliget, Ady u. 36. 2145.

CSERÉLEK

ATARI 800XL típusú személyi számítógéppel rendelkezők ismeretségét keresem. Másolóprogram ellenében szívesen adok játékprogramot és cserélek is. Szkirossy Sándor, Siófok, Mártírok u. 9. 8600.

Commodore 16-os játékprogramokat cserélnék. Váczr Rudolf, Kiskunfélegyháza, Klapka u. 25. 6100.

C64-es játékprogramokat cserélek lemezen. Szabó Géza, Nyíregyháza, Városmajor u. 26. fsz. 1. 4400.

C64-re készült játék- és egyéb programokat cserélnék. Csak lemezen! Mátyási Árpád, Jánoshalma, Kossuth u. 38. 6440.

C64-re programokat cserélek, kazettán. Dóznai fibor, Budapest, Klapka u. 4. II. 6. 1193. Tel.: 770-193.

C64-es programokat cserélek. Garamvölgyi Gábor, Pécs, Ybl Miklós u. 7/3.5. 4. 7633. tel.: 72/13-166.

Commodore 64-re programokat cserélek lemezen és kazettán. Válaszokat listával kérek. Boldizsár Gábor, Kék, Petőfi u. 23. 6446.

Commodore 16-os, 20-as, 64-es 128-as géppel rendelkezőkkel kapcsolatot keresek. Negyvenöt programom van. Gyermán Sándor, Zrenjanin Rade Koncara 23/V. 23000.

ZX-Spectrum (48 k) játék- és felhasználói programokat cserélek. Válaszokat listával erre a címre várok: Kiss Szabolcs, Pécs, Eszék u. 19. 7632.

Videoton TV-Computer tulajdonosok ismeretségét keresem. Sebestyén Pál, Szolnok, Hunyadi u. 44. 5000.

BASIC és gépi kód

Legutóbb az összehasonlító és a feltételbitek értékét állító utasításokról volt szó. Most az aritmetikai utasításokkal ismerkedünk meg.

Az aritmetikai utasítások

Két utasítás tartozik ebbe a csoportba: az összeadó ADC és a kivonó SBC. Néhány szerző a számláló utasításokat is ide sorolja, de ez nem általános.

Az aritmetikai utasítások végrehajtása az A regiszterben történik, az egyik operandusnak az utasítás végrehajtása előtt már a regiszterben kell lennie, az eredmény is ott képződik. Az eredménynek megfelelően az állapotregiszter négy jelzője állítódik be: a C, N, Z és V bitek. A C (átviteli) bit végrehajtás előtti állapota befolyásolja az eredményt, ezért gondoskodni kell a megfelelő érték beállításáról.

A decimális aritmetika használatánál az aritmetikai utasítások működése eltér az itt leírtaktól.

Az ADC utasítás

Az ADC (ADd with Carry = összead átvitellel) utasítás az A regiszter tartalmához hozzáadja az operandus által meghatározott értéket és a C bit aktuális értékét. A Z és N biteket a már korábban megismert módon, az eredménynek megfelelően állítja be.

A C bit értéke akkor lesz 1, ha a legmagasabb helyértéken átvitel képződik, vagyis ha az A regiszter tartalmát előjel nélküli 8 bites számnak tekintve az összeadás eredménye 255-nél nagyobb lesz, és nem fér el 8 biten.

A V bit akkor állítódik 1-re, ha az összeadásnál a regiszter 6. bitjéről a 7. bitre van átvitel. Ennek az A regiszter tartalmának előjeles számként kezelésekor van jelentősége. Ekkor ugyanis az említett átvitel együttal azt is jelzi, hogy a keletkezett eredmény valamilyen irányban átlépte az ábrázolható számtartomány határát, azaz túlcsordult. A bekapcsolt V bit — melynek neve az angol overflow (túlcsordulás) szóból keletkezett — éppen ezt jelzi.

Fontos, hogy az összeadás előtt a C bit értéke megfelelő legyen. Ezt általában a CLC utasítással állítjuk be.

1. lista

```
100 rem rom masolasa ram-ba (c64)
110 for i=828 to 855
120 read a : poke i,a : s=s+a
130 next
140 if s=4661 then sys 828 : end
150 print "adathiba!":stop
828 data 169,160,160,0,32,69,3,169
836 data 224,162,32,132,251,133,252,177
844 data 251,145,251,200,208,249,230,252
852 data 202,208,244,96
```

Az SBC utasítás

Az SBC (SuBtract with Carry = kivon átvitellel) utasítás az A regiszter tartalmából kivonja az operandus által meghatározott értéket és a C bit aktuális értékének ellentettjét. A Z és N biteket az SBC is az eredménynek megfelelően állítja be.

A C és V feltételbitek beállítása hasonló, mint az ADC utasításban, azzal az eltéréssel, hogy ezeknek a biteknek az értéke akkor lesz 1, ha a megfelelő helyeken nincs átvitel. A kivonást követő esetleges feltételvizsgálatnál eszerint kell eljárni.

Fontos, hogy a kivonás előtt a C bit értéke megfelelő legyen. Ezt általában a SEC utasítással állítjuk be.

Még egyszer a LOAD-ról

A decemberi számban írtam arról, hogy a C64-es géppel és mágnesszalagos egységgel dolgozva, egy BASIC "LOAD" parancs hatására a gépi kódú program az eredeti helyére töltődött. Ezzel kapcsolatos Pásthy Béla tatabányai olvasónk hozzászólása, melyből az erre vonatkozó részt idézem: „Nem tudom (. . .) milyen gépe van, de az enyém LOAD parancs hatására az $2B-$2C címtől veszi a kezdőcímet, és függetlenül a fejlécbeli (szalag)információktól, ide tölti be. Mi a magyarázata az ön gépénél a helyes címre töltésnek?"

A januári számban — amely olvasónk levelének érkezése táján jelent meg — röviden már utaltam a megoldásra, de úgy látom, részletesebb magyarázatra van szükség.

Az említett gépi kódú program szalagra írásakor 1-es másodlagos címet használtam. Ekkor a SAVE KERNAL rutin a szalagra kerülő programfájl fejlécébe 3-as típusjelet ír be, ami a betöltésnél a LOAD KERNAL rutinnak jelzi, hogy a programot a fejlécben található címtől kell a tárba tölteni. A C64-en a BASIC LOAD parancs hatására a 3-as típusú fájlok a helyükre töltődnek. A VC20-on ez nem így van, ezért

```
033c  a9a0        lda #$a0
033e  a000        ldy #$00
0340  204503      jsr $0345
0343  a9e0        lda #$e0
0345  a220        ldx #$20
0347  84fb        sty $fb
0349  85fc        sta $fc
034b  b1fb        lda ($fb),y
034d  91fb        sta ($fb),y
034f  c8          iny
0350  d0f9        bne $034b
0352  e6fc        inc $fc
0354  ca          dex
0355  d0f4        bne $034b
0357  60          rts
```

2. lista

— a novemberi µM-ban leírt módon — közvetlenül a KERNAL LOAD rutinjával kell a betöltést elvégezni.

Olvasónk problémája abból adódik, hogy programját nem 1-es másodlagos címmel mentette ki, hanem — feltehetően — a BASIC programmutatók átírása után SAVE "FILENÉV" BASIC paranccsal. C64-en kedvező eredményt érhet el a SAVE "FILENÉV",1, 1, formával. Elkerülheti a BASIC programmutatók átírásával kapcsolatos fáradságot, ha a jelen sorozat 1986. novemberi számában megjelent részében közölt SAVE rutint használja, természetesen a mágnesszalagos tárolásra vonatkozó módosítással.

A témához tartozik még, hogy dr. Úry László Commodore könyveinek a kazettás tárolásról szóló részeiben a 3-as fejtípusról egy szó sem esik, mintha ez nem is létezne.

ROM másolás RAM-ba

Mostani példaprogramunkat csak C64-en lehet kipróbálni. Az indirekt indexelt címzési mód célszerű alkalmazására mutat be gyakorlati példát. A program feladata, hogy a C64-en a beépített BASIC és a KERNAL ROM tartalmát másolja a vele azonos címtartományban lévő RAM-ba. A memória 1-es címén — a processzor-porton — található érték határozza meg, hogy a processzor a ROM-ot vagy RAM-ot látja, vagyis melyikből olvassa ki az információkat. A beírás azonban mindig a RAM-ba lesz.

Az 1-es című bájt tartalma alaphelyzetben — pl. a gép bekapcsolása után — 55 ($37). Ha ezt 54-re ($36) változtatjuk, az interpreter címtartományában ($A000 . . . $BFFF) a processzor a RAM-ból olvas. Ha az 1-es címre 53-at ($35) írunk, az interpreteren kívül az operációs rendszer címtartományában ($E000 . . . $FFF) is a RAM-ot látja a processzor.

A fentiek felhasználásával kisebb módosításokat hajthatunk végre a beépített rutinokban, úgy, hogy az itt közölt program-

mal a ROM tartalmát a RAM-ba másoljuk, majd a szükséges módosítások végrehajtása után az 1-es tárcímre beírjuk a megfelelő értéket.

Jó példa az alkalmazásra a februári szám 34. oldalán közölt VC20 programok szalagról való beolvasására szolgáló program. Mostani programunk az ottani BASIC változat első két sorának feladatát végzi el, kb. 120-szor gyorsabban. Futtatása után az ottani 2. listán látható programot betöltve és RUN 120 parancsal indítva érhetjük el a kívánt eredményt. Ne feledkezzünk meg róla, hogy ez a mostani program csak a másolást végzi el!

A BASIC betöltő az 1. listán látható. A végleges változat elmentése előtt célszerű a 140 . . . 150 sorokat az alábbiak szerint módosítani:

140 sys 828
150 new

A 2. lista disassemblerrel készült. Ezen tanulmányozhatjuk a program működését, és akinek kedve van, összehasonlíthatja a feladatnak a fentebb említett programban alkalmazott megoldásával.

A program magva a $0345 . . . $0357 tartományban elhelyezkedő másoló szubrutin. Ez kétszer fut le, először a $0340-es címen lévő JSR utasítás hatására, másodszor közvetlenül kerül rá a vezérlés.

A szubrutin egy alkalommal 32 memórialapot másol a ROM-ból a RAM-ba. Az X regiszter a számláló. A másolás mutatója (pointere) a $FB . . . $FC című bájtokon van. A másolás azonos címre irányul, de — mint tudjuk — a ROM-ból a RAM-ba. Megfigyelhetjük, hogy a másolás során a mutatónak csak a magasabb helyiértékű — más szóval felső — bájtjának tartalma változik, az alacsonyabb helyiértéken mindig 0 van, a címet indexelés módosítja.

A szubrutin végrehajtása előtt az A regiszterbe töltjük a másolandó terület kezdőcímének felső bájtját, az Y-ba 0 kerül.

Figyeljük meg, hogy az Y-ba csak egyszer viszünk be explicit módon nullát! Gondolkozzunk rajta, hogy miért tehetjük ezt meg!

BARNA LÁSZLÓ

HCC PRIMO szakosztály

A PRIMO gyártásának megszüntetése nem javította a gép tulajdonosainak és felhasználóinak helyzetét. Ez is indokolta, hogy a Neumann János Számítógéptudományi Társaság keretei között megalakuljon a HCC PRIMO szakosztály.

Az összejöveteleket általában minden hónap második hétfőjén 18 órától a COMPANIA-ban (Szabadidő Központ, Budapest VII., Almássy tér 6.) tartják.

BASIC igényesnek, extrákkal eladó . . .

„A rossz, mit ember tesz, tuléli őt,
A jó gyakorta sírba száll vele."
(Shakespeare: Julius Ceasar. III. f. 2. sz.
Ford.: Vörösmarty Mihály)

Bakos Tamás „A programozás Trabantja" c. cikke (µM 1987. 2. szám, 14. oldal) érdekesen mutatja be a BASIC előnyeit, hátrányait. Kár, hogy a vizsgálat a BASIC előnyeinek és hátrányainak összevetésére korlátozódott, más nyelvekkel való összehasonlítás nélkül. A magas szintű nyelvek közül a COBOL és FORTRAN neve egyszer, az ALGOL-é kétszer fordul elő. A cikk szerint a BASIC „. . . túlélte az ALGOL-t . . .„

Ezt a mondatot olvasván óhatatlanul eszünkbe ötlik e hozzászólás mottója. Bizonyíthat-e bármit is a túlélés? Netán azt, hogy a BASIC „jobban sikerült", mint az ALGOL?

Szűklátókörűségre vallana, ha bármelyik magas szintű programozási nyelvet másképp értékelnénk, mint munkaeszközt. Az elvégzendő feladat akár egy játékprogram írása is lehet, bár ez vajmi kevéssé segíti a valós társadalmi-gazdasági gondok megoldását. Abból kell kiindulni, hogy a cél az összes ráfordítás csökkentése. Ebbe beletartozik mind a géptartozék, mind a felhasználói programok létrehozása. A magas szintű nyelvek fordítóprogramját géptípusonként elég egyszer megírni, de *minden új feladathoz új felhasználói programra van szükség.* Akkor, ha a fordítóprogram gyenge, a felhasználók végeznek felesleges programozói munkát; ez lehet a fordítóprogram hibásan működő részeinek hatástalanítása is, „trükkös" programok írásával.

A „fordítóprogram" szó helyére beírhatjuk a „BASIC értelmezőrendszer" kifejezést is, a helyzet nem változik. Lássunk erre néhány gyakorlati példát.

A BASIC *könnyű elsajátíthatósága* csak a matematikai modulra igaz. Az egyes BASIC-változatok fájlkezelésében csak az alapszavak (OPEN, CLOSE, PRINT# stb.) egyeznek, az utasítások hatása eltérő. Ezt még egy-egy gyár különböző géptípusainál sem mindig sikerült elkerülni. (Lásd: Commodore 64 és C16 programok a µM hasábjain.) Figyelembe véve, hogy a mikroszámítógépek manapság 3-4 év alatt elavulnak, ez azt jelenti, hogy a BASIC felhasználói 3-4 évenként átírhatják programjaikat. — És még nem is esett szó a string-modulról, a mátrix-modulról stb. sem!

A BASIC az ALGOL-lal ellentétben, a FORTRAN-hoz hasonlóan, csak a tömbök deklarálását igényli. Ennek következménye, hogy *a program a gépelési hibákat is végrehajtja.* Kivel nem fordult még elő, hogy az „X=Y" helyett „X=Z"-t írt? (Közismert, hogy az „Y" és a „Z" az angol és a német [+magyar] billentyűzeteken fel van cserélve.) Mivel a Z-t ugyanúgy *nem kell deklarálni,* mint az Y-t, a program a nem definiált Z-vel fog dolgozni. A BASIC egyszerűsége megbosszulja magát: egyetlen ilyen programhibával kapcsolatos *bogarászás több időt igényel,* mint amennyire a deklaráció beírásához lenne szükség. (Ráadásul a fordítóprogramnak is egyszerűbb lenne a dolga . . .) Az a tény, hogy az X-et, X%-ot és X$-t meg kell különböztetni, ezen nem segít. Egyébként az „új" FORTRAN, a FORTRAN 77 megkívánja a karakter típusú változók maximális hosszának deklarálását. A felhasználó kényelmére (?) több BASIC rendszer, így a Commodore is eltekint ettől. A következmény jól ismert: ha a felhasználó nem tűzdeli tele programját *üres* X=FRE (0) utasításokkal, egykettőre megtelik a stringterület, és a rendszer befagy.

A BASIC-ből nemcsak az „igazi eljárások" (paraméterátadás) hiányoznak, hanem a blokkstruktúra (lokális változók) is. Feladata válogatja, melyik a súlyosabb hiány. Ha például a programban sokat kell integrálni, egy szummázó eljárás nagyon jól használható. Sajnos a következő (a blokkstruktúrából adódóan rekurzívan is hívható) eljárást a cikkben említett magas szintű nyelvek közül csak ALGOL-ban lehetett megírni:

```
real procedure SZUMMA (VÁLTOZÓ, ETTŐL, EDDIG, LÉPÉS, KIFEJEZÉS);
begin real S;
  S:=0;
  for VÁLTOZÓ:=ETTŐL step LÉPÉS
      until EDDIG do S:=S+kifejezés;
      SZUMMA:=S
end
```

Mivel **value**-l lista nincs az eljárásfejben, az összes paraméter a ciklus összes lefutása során újra és újra kiértékelésre kerül. Így aztán ez az egyszerű eljárás alkalmas akár az integrálandó függvény viselkedése szerint változó lépésközű integrálásra is, ha a LÉPÉS aktuális paraméterét helyesen adjuk meg. (Alkalmazási példa közlése a BASIC-től túl messze vezetne, így el lehet tőle tekinteni.) *A BASIC programozója egy ilyen egyszerű összegzésre is egyedi programot írhat,* s legfeljebb az vigasztalhatja, hogy a FORTRAN programozója sem tehet másképp.

A nem a hobbi kategóriába tartozó mikroszámítógépek alapprogramjai közé tartoznak a különféle szerkesztő (editor) programok. Mindössze 1-2 perc többlet várakozási idő árán egy megszerkesztett, mondjuk FORTRAN programot is lehet fordíttatni. A fordítóprogram (és esetleg a betöltő program) üzenetei a teljes felhasználói programot minősítik. Nem fordulhat elő, hogy egy hosszú ideig dolgozó FORTRAN program utolsó sorában lévő szintaktikus hiba miatt az egész munka veszendőbe menjen — ami viszont a BASIC értelmezőrendszerek többségénél megtörténhet.

A BASIC által nyújtott *szerkesztési lehe-*

tőség tehát nem tudja ellensúlyozni azt a körülményt, hogy a feldolgozás soronként történik (kedvező esetben a beírással egyidejűleg).

Ritka kivételként említést érdemel a Központi Fizikai Kutató Intézet INTEL-8080 alapú ICC gépének DOS-80 BASIC-je: a billentyűzéskor vagy az előre megszerkesztett program háttértárolóról való beolvasásakor történő soronkénti ellenőrzésen kívül a RUN parancsra az indítás előtt az egész programot szintaktikusan ellenőrzi (például ugrási címek megléte?); kívánság szerint a forrásprogramon kívül megőrzi a lefordított programot is. Érthető, hogy ez a rendszer ennyire alapos: a CAMAC folyamatszabályozó perifériarendszert vezérli — a nem egyértelmű feldolgozásnak az ipari gyakorlatban lennének súlyos következményei.

A PEEK-POKE utasításpár bevezetése *bizonyítja a mikroszámítógépek BASIC alapú fejlesztésének teljes csődjét. Az IBM DOS FORTRAN-IV* már húsz évvel ezelőtt felkínálta a programozóknak az operációs rendszer különleges szolgáltatásait egy egyszerű CALL SYS utasítás révén. *Ne a felhasználó gondja legyen* annak nyilvántartása, hogy az operációs rendszer (például KERNAL) változói, szubrutinjai milyen címen találhatók, hogy azután decimálisan kódolt hexadecimális adatokat (esetleg „fordított" sorrendben) a POKE utasításokkal beírhasson, ha a rendszer valamilyen szolgáltatását igénybe akarja venni. Ha pedig a feladat másképp nem oldható meg — kár kísérletezni a BASIC-kel, gyorsabb, rövidebb, elegánsabb programot lehet írni gépi kódban. Az olvasók gyakran találkozhatnak programokkal, melyeknek mind az assembly, mind a BASIC listáját közli a µM; ezek különbsége szembetűnő. *Ellensúlyozzák a BASIC előnyei a felhasználó többletmunkáját? Indokolt a BASIC használata más nyelvek helyett?*

A hozzászólás mottójához visszatérve: ezúttal is a költőnek lett igaza. Az ALGOL nem (valóban létező) korlátai miatt került ki a használatból, hanem azért, mert *nem állott mögötte egyetlen igazán nagy számítógépgyár sem.* Néhány lelkes programozó nem képes elvégezni azt a munkát, amit a tőkeerős gyárak nem támogatnak. (Ismeretes, hogy a µM-ban is bemutatott Razdan-3 géphez az Egyetemi Számítóközpont — jelenleg Eötvös Lóránd Tudományegyetem SZK — matematikusai készítettek egy igen jó, korlátozásoktól csaknem mentes ALGOL fordítóprogramot, amely szerkesztési lehetőségeket is kínált.) A FORTRAN mögött áll az IBM, a CDC, a DEC stb.; a BASIC-et támogatja az Apple, a Commodore stb. — az európai eredetű ALGOL kiszorult a világpiacról.

Ha *gyorsan és pontosan* kell számolni, egy HP-97 asztali kalkulátorral (aminek kiterjedt programkönyvtára is van) hamarabb kaphatunk eredményt, mint a BASIC DATA-READ sorpárainak feltöltésével; a 12 decimális jeggyel dolgozó, verem rendszerű processzor pontossága az igényeket szinte mindig kielégíti.

Ha *bonyolult számításokra van szükség*, kész eljárásokat találhatunk a FORTRAN könyvtárakban (például IBM FORTRAN-IV SSP-Scientific Subroutine Package). Nincs gond a változók átnevezésével stb.

Ha *ügyviteli* feladatot kell megoldani, látványos táblázatokkal: rendelkezésre áll a COBOL, de a FORTRAN-ban való programozás sem okoz gondot.

Voltaképp *nehéz olyan területet találni, ahol éppen a BASIC lenne hatékony.* Ha pedig valamelyik gépi megvalósítása sok többlet szolgáltatást kínál, az pont olyan, mint a hirdetésekből jól ismert *„Trabant, igényesnek, extrákkal . . .":* kétségtelen, hogy megy, de sokat fogyaszt (programozási munkát) és szennyezi a környezetet (erőltetett megoldásokkal).

Ahogy a hasonlatoktól elvárjuk, a Trabant-BASIC hasonlat is sántít. *Túlértékeli* a BASIC-et. Ha valaki Trabanton szerzi meg a jogosítványt, nyugodtan vezethet Rolls Royce-ot is. A BASIC-vizsgát tett új programozó nemhogy az ALGOL, de még a FORTRAN programokat sem érti meg. Remélhetjük, hogy a felhasználói szoftver fejlesztése lépést tud tartani a hardverlehetőségek bővülésével? *Félő, hogy nem.*

SZONDI EGON JÁNOS

. . . S mit szól mindehhez a rovatszerkesztő, Bakos Tamás? Adjuk át neki a szót!

Először is köszönöm, hogy több szempontból kiegészítette szerény cikkemet. En a BASIC-et valóban önmagában akartam értékelni, nem állítottam, hogy az ALGOL-nál jobban sikerült. A túlélés azonban tény, és egy dolgot mindenképpen jelent: a tömegek ítéletét. Nem hiszek abban, hogy egy nyelvet bármelyik számítógépgyártó huzamosan népszerűvé tud tenni. A PL/I nevű öszvér mögött hiába áll(t) az IBM, nem lett belőle telivér: hasonló az Ada és a Pentagon esete is.

Az ALGOL azért halott, mert csak elenyésző kisebbség számára volt érthető; az olyan általános eljárásokat, mint a hozzászólásban szereplő SZUMMA, a név és érték szerinti paraméterátadást, a rekurziót nem a fogyasztói társadalom átlagemberének találták ki. Ugyanakkor a mikrogépek túlnyomó többségét ennek az átlagembernek adják el.

Egyetértek azzal, hogy BASIC-re nem lehet mikrogépes szoftverfejlesztést alapozni, de az tagadhatatlan, hogy az amatőr BASIC programozók hozzájárulnak a számítástechnika társadalmasításához.

A FORTRAN szubrutinkönyvtárak használnát és a COBOL lehetőségeit senki nem vonja kétségbe, de az ezekhez hozzáférők száma elhanyagolható a személyi gépeken „programozók" számához képest.

A BASIC egy főiskolán, oktatási célra készült; karrierjét nem tehetős vállalatok, hanem az utca embere alakította. A sok nyelvjárás a népszerűség átka. Vannak jó és rossz BASIC-rendszerek. A hobbi gépek kategóriájában kiemelkedik a Sinclair BASIC-je, és jogos szidást érdemel a Commodore-é. Ez azonban nem nagyon zavarja a kezdőket, hiszen egy újszülöttnek minden vicc új.

Akit pedig „megcsap a mozdony füstje", úgyis lehagyja a BASIC-et, és megtalálja a neki megfelelő „igazi" nyelvet.

Végül pedig: ha valaki Rolls Royce-ot akar vezetni, ahhoz nemcsak jogosítvány kell, hanem Rolls Royce is.

Utóirat: ha mottót írtam volna, így kezdődne a válaszom:

„Hej, ha ezt az én galambocskám, ALGOL FORTRANOVICS megélhette volna!" (Alfonzó után szabadon.)

Köszönjük Szondi Egon János színvonalas hozzászólását. Ilyen jellegű cikkeket a Magazin máskor is szívesen közöl. (A szerk.)

Kritika és önkritika

Az 1987/1. szám 15. oldalán közölt rendezésről szóló cikk sok hibát és következetlenséget tartalmaz. Nyilván gépelési elírás a program 30-as és 60-as sora, a helyes utasítások:

```
30 MIN=F(X,K):Z=X
60 IF Z=X THEN 80
```

Egy kétdimenziós tömböt rekordokból, a rekordokat pedig mezőkből állónak ugyan el lehet képzelni, de felesleges, ráadásul terminológiai zavarokat is okozhat. A szöveges magyarázat világos, a program — eltekintve a gépelési hibáktól — jól áttekinthető, ezt egy folyamatábrával csak elrontani lehet. Valóban, az első elágazás *igen* ága a levegőben lóg, nincs folytatása, az elemek cseréjét viszont végtelen ciklussal megoldani kissé sokáig tartana. Indokolatlan továbbá az elágazások *igen* ágát a megszokott *i* helyett *j*-vel jelölni.

Maga a cikk témája — hogy az egyszerű rendezések közül a közvetlen kiválasztás módszere a legjobb — közismert, szinte minden, rendezéssel foglalkozó könyvben szerepel (például N. Wirth: Algoritmusok + Adatok = Programok, Műszaki Könyvkiadó, 1982). Mindezek után nagyon furcsa számomra, hogy a jelenlegi ismereteink szerint legjobb gyorsrendezőt és a szintén kiváló Shell-Metzner rendezőt szinte eldugják a 45. oldal sarkában.

Ugyanennek a számnak a 16. oldalán szerepel a bináris keresés BASIC és assembly nyelvű programja. Mindkettő korrekt, viszont nem túl elegáns és egyáltalán nem didaktikus a ciklusból való kilépésre egy, a program szerkezetéből nem következő feltételt használni. Az alábbi BASIC program áttekinthetőbb és hatékonyabb változata a cikkben szereplőnek.

```
10 INPUT "A KERESETT SZO";N%
20 AH%=1:FH%=M%
30 IF AH%>FH% THEN PRINT "NINCS
   ILYEN":GOTO 10
40 K%=(AH%+FH%)/2
50 IF N$(K%)<N$(0) THEN AH%=
   K%+1:GOTO 30
60 IF N$(K%)>N$(0) THEN FH%=K%-1:
   GOTO 30
70 PRINT K%,N$(K%):GOTO 10
```

A ciklus során a még szóba jöhető értékek alsó határának indexe (AH%) állandóan nő, a felső határ indexe (FH%) állandóan fogy, így a keresés eredménytelenségét az jelzi, ha AH%>FH%. E javítások alapján az assembly program is egyszerűsödhet.

Az 1987/2. szám 42. oldalán megjelent írásomban szereplő interpoláló programot Spectrumon készítettem, s kihasználtam azt a ciklusok előtesztelésében következő tulajdonságát, hogy ha a ciklusváltozó kezdőértéke nagyobb, mint a végérték (pozitív növekménynél), akkor a ciklus nem hajtódik végre. Hátultesztelő nyelvekben ezek az „üres" ciklusok is legalább egyszer lefutnak, így például a Commodore 64 BASIC-jében. Hátultesztelő nyelv használatakor tehát a program 80-as és 150-es sorában kezdődő belső ciklusok első, „üres" lefutását át kell valamilyen módon lépni.

LOVRICS LÁSZLÓ

μ'87

A Neumann János Számítógéptudományi Társaság idén is megrendezte az **Országos Mikroszámítógépes Találkozót**, mely nemcsak mint számítástechnikai kiállítás, hanem az informatika és mikroszámítógépek iránt érdeklődők valódi fesztiváljaként is sikeres volt. Megvalósult a rendezők azon szándéka, hogy a látogatóknak a látványon kívül egyéb tartalmas szórakozást is nyújtsanak, és itt nem állt: „mindent a szemnek, semmit a kéznek".

A GÉP-TOTÓN-n bárki kipróbálhatta BASIC programozási ismeretét, sőt egy kis szerencsével apróbb ajándékot nyerhetett. A sakkjátékosok egész nap kihívhatták a világ egyik legjobb (Fidelity-Avantgarde) sakkszámítógépét. A bridzsrajongók számítógép ellen licitálhattak.

A kiállítók pavilonjaiban a szorgalmasak idegen nyelveket tanulhattak; akit érdekelt, kapcsolatba kerülhetett a beszédszintetizátorral egybeépített számítógéppel, és sokan „csak" egyszerűen programoztak.

A hobbiszámítógép-építők és -programozók sok érdekes információval gazdagodtak a HCC-szakosztályok tagjaival való nyílt, közvetlen hangulatú beszélgetések során. E légkörben bizonyára jó néhányan elhatározták, hogy belépnek a Computer Club tagjai sorába.

A chip-cserebere idén nem vált be igazán, mert ki-ki más-más időben jelent meg portékájával, és így nem jött össze az „üzlet".

Akik számítástechnikai ismereteiket akarták bővíteni, érdekes előadások és bemutatók között válogathattak. Nem mindennapi élményt nyújtott például az **Informatika Története és a Digit-Art** kiállítás.

Az egy hétig tartó eseménysorozatról a helyszínen tájékoztatott az **Újdonság Újság**, amelyben csak az nem volt „újdonság", hogy rögtön hiánycikké vált.

Megnyitó

A teljes programból néhány mozaikot emelünk ki, és megszólaltatunk egyes résztvevőket.

A találkozó egyik legértékesebb és igen érdekes programja volt a *Háziépítésű Számítógépek*, az *Oktatóprogramok* és a *Számítógép a családban* országos versenye, vetélkedője. A versenyek külső szemlélő számára is nagy élményt jelentettek, de megkértük Horváth Lászlót, a versenyek zsűrijének titkárát, hogy áttekintve az előzményeket is, adjon értékelést a szakember szemével.

A versenyen mindenki nyert

A *Háziépítésű Számítógépek* országos versenyére — mondta Horváth László — 15 pályamunka érkezett, és 13 résztvevővel zajlott. A zsűri az alábbi szempontok szerint minősítette a pályamunkákat:

— mennyiben önálló tervezésű, azaz értékeltük az alkotó kreativitását;

— a megvalósítás kapcsán vizsgáltuk a műszaki kivitelezés, építés szabályait, és hogy esztétikus-e;

— milyen újszerűséget, ötleteket tartalmaz;

— hogyan kapcsolódik a gépekhez, perifériákhoz, azaz érvényesül-e a rendszerbe való illeszkedés.

Kiemelnék néhány kimagasló pályamunkát. **Sánta Mihály** *moduláris perifériacsaládja* zseniális ötlet. Felhívtuk rá az Ipari Minisztérium figyelmét is. A gépre kapcsolt, egyénileg tervezett perifériák vizuálisan szemléltetik az ott lejátszódó folyamatokat. A különböző modulok a gép egyes egységeinek aktuális állapotát mutatják, így a gyerekek szó szerint szemmel követhetik a számítógép működését. Ennek az ismeretterjesztésben van hatalmas szerepe. Az alkotó, Sánta Mihály a Kiváló Ötlet díjat nyerte.

Urbán Kornél és **Urbán Zoltán** Z80-as *számítógépe* gondosan és esztétikusan elkészített konstrukció. Munkásságuk példamutató az összes iskola számára. A szakma ezt várná el mindenhol a számítástechnika oktatásában. Külön kiemelem a géphez készített FORTH fordítóprogramot. Ha a ver-

„Az informatika története" kiállítás

senyen fordítóprogram-kategória is lett volna, bizonyára azon is jó helyezést érnek el.

Németh László és **Horváth István** *helyi hálózata* kapta a legmagasabb „újszerűség" pontot. Csodálatos a két tanár erőfeszítése, amellyel bebizonyították, hogy kisgépekre is egyszerűen, olcsón lehet hálózatot telepíteni. Ha módszerük elterjedne, nagy előrelépést hozna az oktatás legkülönbözőbb területein. Gondoljunk csak arra, hogy míg a tanár például az osztálykönyvbe ír, addig is minden gyerek bizonyíthat: egyszerre „felelnek", és a tanár megkapja a kinyomtatott eredményeket. Minden gyerek biztosan felel. A páros alkotása a szoftverek között is nyert: programjuk javít, mér és oktat.

A zsűri nekik ítélte a III. különdíjat is, azonban a TII ezt nem hagyta jóvá. Más szempont alapján értékelt — a legtöbb pályamunkát beküldő iskolákat kívántak jutalmazni. Nem hiszem, hogy ilyen típusú versenyeken a mennyiség számítana. Szerintem a fonyódiaknak meg kellett volna a díjat kapniuk. (A μSzámítógép Magazin felajánlotta a TII helyett a megítélt díjat, és a versenyzőknek el is juttatta. — A szerk.) A többi résztvevő is dicsérhető. A verseny színvonalával elégedett vagyok, jövőre sem kell változtatni a lebonyolításán.

Az *oktatóprogramok versenyére* 109 nevezés érkezett. A pályamunkák nagy száma miatt a korrekt elbírálás érdekében a versenyt megelőzően három napon keresztül előzsűrizés folyt. 74 programot tartottunk elbírálhatónak. A kizárás okaként főleg azt említhetem, hogy ezek nem oktatóprogramok voltak.

A bemutatás nyolc kategóriában, két zsűri előtt zajlott. Meglepetést okozott a biológia—kémia kategóriában indult versenyzők jelentős aránya. E programok átlagosan is jók, érdekesek voltak, s néhányuk komolyan kihasználta a gép grafikai lehetőségeit is. **Demeter László** *Növények anyagforgalma* c. programja olyan grafikát produkált, melyet iskolaszámítógépen csak nagy türelmet igénylő munkával lehetett előállítani. Hasonlóan igényes kidolgozású **Schweighoffer Edina** és **Tamás** munkája is. Mindhárman első díjat nyertek, ami a SOTE-t is dicséri.

A matematikától mindenki többet várt. Ha összehasonlítható lenne, azt mondanám, hogy a többi kategória köröket vert rá. Első díjat nem is adtunk ki.

Kellemes élményt nyújtott **Balogh Csaba** *Hullámmozgás* és **Mészáros Tamás Csaba** *Felületi hullámok* c. pályamunkája. Ötletesen és gyönyörű grafikával szemlélteték a fizikai jelenségeket. A fizika kategóriában ők egy debreceni iskolából két első díjat vittek el.

Helyi hálózat

A számítástechnika tanítását jelentősen segíti **Nagy Imrének** a *számítógép működését* bemutató programja. Ő valóban fel is használja ezt napi tanári munkájában.

Itt jegyzem meg, hogy mindkét versenyben az első díjak 2-3 iskolához kapcsolódnak. Ez azt mutatja, hogy ezek az élenjáró iskolák nagyon komolyan veszik az informatika bevezetését az oktatásba.

Különlegességként mutattak be PC gépen egy, az Assembly nyelv elsajátítását támogató általános programcsomagot, amelyet **Diószegi Zoltán** és **Csitári Gyula** készített. A program lépésről lépésre haladva ismerteti meg az MS—DOS-t és az Assembly programozási nyelvet. A munka jelenleg még minta, de bővíthető. Figyelemmel kellett lenni arra, hogy más gép más szemléletet igényel.

Sajnos a humán kategóriában gyenge teljesítményt láttunk. Ezt azért is nagyon sajnálom, mert több érdeklődő vallotta be csalódását, akik csak azért jöttek ki a rendezvényre, hogy megismerjenek politikai gazdaságtan, történelem, irodalom tárgyú oktatóprogramokat.

A fődíj megítélésekor a zsűri rendkívül alaposan mérlegelt, hiszen az erkölcsi megbecsülésen túl értékben is nagyságrendi a különbség a többi díjhoz képest.

Gálai Antal, dr. Medgyes Péter, dr. Albert Sándor, Knáb Erzsébet LINGUASOFT

Rák ellen – az emberért, a holnapért

A μ'87 Kiállításon a rák elleni küzdelmet támogató társadalmi alapítvány is megjelent. Hogyan és miért kerültek a kiállításra? A válasz egyszerű: a betegségek megelőzésében, gyógyításában mindinkább teret nyer az informatika. A szakosodás elterjedésével, a tudományos eredmények növekedésével ugyanis az informatika felhasználása nélkül nem lehet lépést tartani. Így az információ közvetlenül elősegíti a gyógyítást.

Ez a társadalmi alapítvány hozta létre — és a μ'87 Kiállításon be is mutatták — az IBM PC kompatibilis gépre szervezett Magyar Onkológiai Bibliográfiát. A katalógus adatbázisát az 1975-től 1985-ig terjedő, a rákkal kapcsolatban megjelenő könyvek, cikkek, előadásanyagok adják. Az adatbázist folyamatosan bővítik.

A bibliográfia segíti a szakképzést, a kutatást és a gyakorló orvost, akinek szinte megoldhatatlan a szakirodalom figyelemmel kísérése. A számítógépes katalógus által megkérdezheti, milyen anyag van az őt érdeklő témában:

— Szerző
— A betegségek nemzetközi kódja
— Tárgyszó
— Időintervallum
— Kiadványok

szerint, vagy ezek variációival. Például, hogy egy szerzőtől adott időben és valamely betegségről milyen cikkek, illetve könyvek jelentek meg. A mikroszámítógépek remélhető gyors elterjedésével bármelyik kórházban, rendelőintézetben az orvos megtudakolhatja, hol talál adatot, illetőleg részletesebb tájékoztatást az éppen felmerült problémáira.

Az alapítvány célja, hogy a számítástechnikai alkalmazást kiterjessze a betegség megelőzésére, diagnosztizálására és kezelésére.

Lapunk készségesen ad helyt a toborzásnak: Kérik a csatlakozni kívánókat, akik társadalmi munkában tudásukat, szabadidejüket adnák az alapítvány célkitűzéseihez, **jelentkezzenek!** A kezdeményezéseket, a személyi felajánlásokat az Állami Ifjúsági Bizottság Titkárságán (Bp. XIII., Hegedűs Gy. u. 79/81. Tel.: 497-355) várják.

oktatóprogramja elökészítésében, megalkotásában egyedülálló volt. A LINGUASOFT kiválóan edzi a tanulni vágyó fordítási készségét, ezenkívül aktívan ellenőrzi is tudását. A tanár bármely időpontban lekérdezheti a gépet, hogy melyik tanulónak milyen kérdéskör okoz problémát, mert a próbálkozásokat memóriájában megőrzi. Nagyszerű leleménye a program készítőinek, hogy saját maguk is követik termékeiket: a felhasználónak kb. fél év után hajlékonylemezt küldenek, mely „lekérdezi" az addigi tapasztalatokat. Az információ birtokában a program fejleszthető, módosítható.

Megemlítem, hogy a KISZ KB, a KSH és az Ipari Minisztérium ajánlott fel különdíjat, de rajtuk kívül két magánszemély is. Ők tavaly a HCC-versenyen helyezést értek el, és úgy érezték, hogy támogatniuk kell az idén indulókat. Szép gesztus volt.

A KISZ KB 10 000 forintos különdíját nem egy összegben adtuk ki, hanem az arra érdemeseknek könyvjutalmat osztottunk. Így minden kissrác szép könyvvel tért haza.

A *Számítógép a családban* vetélkedőn 9 „csapat" indult. Nem hiszem, hogy a valóságban ennyire kevés családot érintene csak a számítógép otthoni jelenléte. És hogy ne legyen család, ahol például a fűtést, világítást számítógéppel vezérlik . . .

Szabó Géza és családja hat főt számol. Náluk teljesen igaz, hogy a számítógép mindenkié. A papa elkészíti a gyerekek tanulási programjait, ők ennek segítségével készülnek fel az órákra. Még a karonülő kisbaba is nyomkodta a billentyűket: a gép villogott és egy lepke röpködött. A kisgyereknek mindez persze nagyon tetszett.

Schermann Gáborék a könyveiket tartják nyilván és a pénzügyi „folyamataikat" vezetik a gépen. Meg kell mondani, hogy az ezen a vetélkedőn szereplő programok elég gyengécskék voltak, de a verseny célja szerint itt az abszolút amatőrök indultak. A nyertes családi program az oktatási kategóriában például nem kapott volna helyezést.

A verseny rendezői továbbra is azt szeretnék, hogy mondjuk a papa, aki a számítógép bolondja, és ezért sok szidást kap, védelemben részesüljön. Ennek legjobb útja, hogy a családban a többiek is minél szélesebb érdeklődési körben „varázsolódjanak el" a számítástechnikától.

Most pedig következzenek a nyertesek.

Mindenki minden órán felel

Horváth István és **Németh László**, a fonyódi Karikás Frigyes Gimnázium és Szakközépiskola tanárai PTK 1096 számítógéppel kezdték a nebulók oktatását, és mára szinte egyedülálló laboratóriumot hoztak létre iskolájuk kollégiumában.

— Miben állt az ötlet? — kérdeztem az alkotókat.

— Az országos programnak megfelelően iskolánkba érkeztek a HT—1080Z számítógépek. Rögtön törni kezdtük a fejünket, hogyan tudjuk azokat legjobban hasznosítani. Célunk nem csupán a számítástechnika megismertetése volt, hanem hogy a többi tantárgy elsajátítását is segítsék elő a masinák. Iskolánkban megalakult a Számítástechnikai Munkaközösség, melynek hét tagját nem kizárólag mérnök-matematika tanárok teszik. A jelenleg alkalmazott helyi hálózat kitalálása és megépítése hosszadalmas folyamat volt ugyan, de azért lett végül eredményes, mert mindvégig törekedtünk az azonos típusú gépek „megszerzésére". Éreztük, hogy ezek helyi hálózatba építése lesz az igazi előrelépés a számítástechnika és a többi tantárgy oktatásában.

(Megjegyezzük, hogy a nyilatkozók nagyon szerényen viszonyultak munkájukhoz, szinte úgy kellett kihúzni belőlük teljesítményük néhány szavas értékelését.)

— Első próbálkozásunk arra irányult, hogy egyik gépről a másikra programokat adjunk át. Azonban a magnóbemenetek nem adtak elég erős jelet, és lassú is volt az adatátvitel. Az erősítő közbeiktatása pedig nagyon drága. Ekkor kezdtünk el gondolkodni azon, hogy a porthoz alakítsuk ki hálózatot.

A helyi hálózat hardverigénye minimális. A HT két port csatlakozójáról tizeres, a kereskedelemben kapható, árnyékolt kábellel kötjük össze a gépeket. Nyolc bit az adatforgalom bonyolítására, két bit a vezérlésre, szinkronizálásra, azonosításra és visszajelzésre szolgál. A gépeket sínrendszerben kötjük össze. Jelenleg 36 gép lehet a hálózatban, de ez változtatható. A kimeneten szabadon maradó 6 bit a gép azonosítását szolgálja, de ezt magába a gép csatlakozójába építjük be. A helyi hálózat egyéb feltételeit szoftver biztosítja, így a gépeket egyáltalán nem kell átalakítani. A hálózat élesztéséhez a központi gépen a tanár kazettáról vagy lemezről beviszi a hálózatszervező szoftvert. A tanulóknak is be kell

LINGUASOFT

saját gépükbe tölteniük egy kicsi programot, ezáltal válik gépük a hálózat részévé, egyébként önálló gépként működik. Az adatátviteli sebesség 10 000 karakter/s.

Mivel a hálózatot szoftver szervezi, több működési variáció is elképzelhető: például ilyen a vizsgáztatás, visszakérdezés. Ekkor a tanár a központi gépen keresztül kiküldi a programot. A tanulók saját gépükön válaszolnak a kérdésekre. A tanár folyamatosan figyelemmel kíséri, ki hol tart, és végül az osztály eredményeit név szerint kinyomtatja. Máskor a központ oktatóprogramot küld ki a tanulóknak, és ők ezzel dolgoznak. A tanár figyeli, hogy ki hol tart, melyik téma okozza a problémát. Mivel az ő gépe bármelyik géppel kapcsolatot teremt, a megfigyelés tetszőleges sorrendben haladhat, egyes diákokat többször is ellenőrizhet. A harmadik variáció szerint a gyerekek önálló feladaton dolgoznak. Ekkor ők kezdeményeznek: vagy segítséget kérnek a központi géptől (a tanártól), vagy megküldik a kész eredményt. Előfordulhat,

Friss hír – nonstop szerkesztőség

A µ'87 Kiállítás egyik standján állandó nyüzsgés volt. Emberek rohangáltak, nyúzták a „védtelen" számítógépeket, a fénymásoló szinte izzott. Itt készült az Újdonság Újság. Az SZKI — SCI'L a µ'87-re kihelyezett szerkesztőséggel lepte meg a látogatókat. Újságíró gyakornokok begyűjtötték a kiállítás híreit, és szinte rögtön el is készült a cikk. Minek köszönhették ezt a gyorsaságot? A PROPER 16 W számítógépeknek és a PROPRESS kiadványszerkesztő programnak.

Az újságírók bebillentyűzték cikkeiket, utána rögtön korrigálták. A többit elvégezte a számítástechnika. A szöveg megkapta tördelt alakját, majd az „újságszerkesztő" gépen a három PROPER által átadott anyagból összeállt a kiadvány, az Újdonság Újság. Mindezt néhány utasítás hatására a PROPRESS állította elő. Az „okos" kiadványszerkesztő számára a grafika, a matematikai képletek, a tipográfia nem okozott problémát. A görög abécé betűin sem akadt fenn. Végül a mátrixnyomtató szorgosan sorolta az SZKI—SCI'L Újdonság Újság oldalait, amelyről készült fénymásolatot frissiben kaphattak a frissességre kíváncsiak.

hogy a diákok programozni tanulnak és valamelyikük „műve" nem működik. Átküldheti a tanárnak a programot, ő belejavít és visszaküldi a helyes megoldást. Nagyobb terjedelmű programot szegmensenként is megcsinálhatnak, és a központi gép „egybefésüli" azt.

— Hogyan jelentkezik be a tanuló a tanárhoz?

— A tanulók gépei alaphelyzetben vételre állnak. A központi gép azonban folyamatosan lekérdezi azokat. Ha a diák kapcsolatot akar teremteni, adásra állítja gépét. A tanár ezt rögtön észleli.

— Mi történik, ha a diák összevissza kezd el gépelni?

— A központi gép programját nem tudja elrontani, de még bejelentkezésnek sem számít az ilyesmi. A központi gép csak „értelmes" információt fogad el. A hálózat által megvalósul többek között az abszolút differenciált oktatás. Az oktatóprogrammal dolgozva ugyanis mindenki a saját tudásszintjén tanulhat, és nem kell alkalmazkodnia a másik, esetleg gyengébb képességű társához.

A gépeket az iskola kollégiumában helyeztük el, ahol este tíz óráig bármelyik tanuló gyakorolhat.

Mikor elköszöntem, Horváth István átadta PTK 1096 számítógépen nyomtatott névjegyét, és meghívott iskolájukba egy kis helyszíni tájékoztatásra. Ott leszünk . . .

A konstruktőrök

Urbán Kornél és unokaöccse, **Zoltán**, szintén kiemelt díjat kaptak. Kornél villamosmérnök, a Landler Jenő Szakközépiskolában vezeti be diákjait a számítástechnika rejtelmeibe; Zoltán ugyanott negyedikes. A „Hogyan is jutottak el a számítógép építéséig?" kérdésre a történetet számunkra így foglalták össze:

— Az iskolában a számítástechnikai mellett beindult a technikusképzés is. A téma iránt érdeklődők részt vesznek az iskolában működő klub foglalkozásain. A számítógép is ennek keretein belül épült meg. A saját tervezésű és készítésű számítógép folyamatos hardverépítés és kísérletezés eredménye.

Először mozaikszerűen készítettünk el egy gépet. Az egyes áramköröket külön-külön alaplapon csináltuk meg, és ezeket összekapcsoltuk. A metódus sok hibát rejtett magában, és többletköltséggel is járt. Ezután terveztük meg az egybeépített gép alaplemezét és kezdtük el az építést.

— Számomra elég különösnek tűnik, hogy szakközépiskolában ilyen szintre eljussanak.

— Az említett klub valójában hardverépítő szakkör. Itt nagyon kreatív társaság jött össze, és szinte követelik, hogy komoly feladatokat kapjanak. Igényük befolyással van a mérnöktanárra is. Így kölcsönös a hatás: sőt egyes területeken a diák, aki rendkívül fogékony, többet produkálhat a tanárnál is. A csoportmunka előnye így teljesen ki tud bontakozni.

A gépépítésnek több területen van jelentősége a képzésben. Először is a hardveres tudnivalókban. Praktikus feladat például a hibásan beültetett alkatrészek megkeresése, a logikai áramkörök ellenőrzése. A másik és egyáltalán nem mellékes eredmény, hogy működő számítógépet állítunk elő, melyet aztán hasznosítunk a különböző tantárgyakban.

— Milyen terveik vannak?

— Megcsináljuk a hajlékonylemez-illesztőt, kialakítjuk a finomgrafikát, azaz a gépet tökéletesítjük. Egyébként a gép jellemzői:

Z80 alapú
64 k RAM
12 k EPROM
2 k video-RAM
2CTC és PU
FORTH értelmezőprogram

— Számítottak-e a díjra?

— Jó eredményre számítottunk, és a Kiváló Oktatásért díj nagy erkölcsi sikere iskolánknak, a szakkörnek külön is. Megerősített minket abban, hogy helyes úton járunk.

„Árulkodó Júdás"

A 25. pavilon galériájára elég nehéz volt feljutnia a látogatónak. A feljáratnál mutatták be a LINGUASOFT nyelvoktató rendszert, amelyet az érdeklődőktől nem volt könnyű megközelíteni. A standon elhelyezett gépeken a magyarról francia, német és angol nyelvre fordítást gyakorolhatták az élelmesek. Valahogy utat törtem **Gálai Antalhoz**, a fődíjas oktató programcsomag szerzőjéhez.

— A műszaki egyetemen építőmérnöknek készültem — mondta Gálai Antal —, mikor már az egész környezetemnek feltűnt, hogy mennyire nem szeretem a felesleges munkát. A matematikafeladatokkal nem szívesen foglalkoztam, mert nem szerettem számolni. Lustaságból, amit csak lehetett, számítógéppel oldottam meg. Kezdeményezőkészségemről és nyugtalanságomról csak annyit, hogy egyszerre két egyetemre szerettem volna járni. Ez nem sikerült, mert nem engedték, de a műszaki egyetem befejezése után elvégeztem Szegeden a matematikus szakot is.

— Hogyan jutott eszébe nyelvoktató programot készíteni?

— Munkám során egyre inkább szükségét éreztem idegen nyelvű szakmai folyóiratok olvasásának, konferenciákon való részvételnek. Ezért 1982-ben beiratkoztam a vállalatom által szervezett angol nyelvtanfolyamra. A csoportban nagy volt a tudáskülönbség. Rengeteget tanultam, és napi három órát hallgattam angol nyelvű rádiót. Sikerült egy-másfél év alatt felzárkóznom, de a magyarról angolra való fordítás mindig nehézségeket okozott. A tanárnak nincs ideje az órán egyénileg foglalkozni a tanulókkal.

Ekkor felismertem, hogy a tanár tudását valahogyan mégis ki kell aknázni. Rögtön elhatároztam, hogy ezt a tudást számítógépbe táplálom, ahonnan bármikor és akárhányszor lehívhatom. 1984-ben angolul jól tudó barátommal kezdtünk nagygépen fordításokat programozni. Elgondolásom szerint a program a tanár-diák viszonyt szimulálva felügyel, feladatot ad, javít és segít. A nagygép a fordításokra csak „jó" és „nem jó" válaszokat adott. Ezután hozzájutottam egy kisgéphez. A program alapelgondolásait megvalósítottam, de a tananyag összeállításához nyelvészekre van szükség. Sikerült megnyernem az ügynek az ELTE-ről **dr. Medgyes Pétert** és **Knáb Erzsébetet**, továbbá a JATE-ről **dr. Albert Sándort**. Ők állították össze a LINGUASOFT nyelvi anyagát.

Az oktatóprogram az egészen kezdő szinttől a középfokú nyelvtudásig használható. A magyar példamondatokat a nyelvészek által előírt véletlen eloszlás szerint adja, mire válaszul be kell gépelni a helyes megoldást. A hibát a gép azonnal, már a szó begépelésénél jelzi. Négy próbálkozásra van lehetőség, illetve előre beállítható időn belül kell a megoldást közölni. A program kérésre végül kiírja az összes lehetséges megoldást.

Mivel a példamondatnak több helyes megoldása van, ezért a feladatot tartalmazó fájlhoz kapcsolódik egy szintaxis rész és egy szótár is, amelyek kapcsolatban állnak egymással. Így nem fordulhat elő, hogy a gép egy helyes, de nem szokványos megoldást nem fogad el. Jelenleg a szoftver 100 órás nyelvi anyagot képvisel. Megfigyelésünk szerint ennek ötszöri ismétlése biztos tudást eredményez.

Az oktatóprogram egyben „árulkodó Júdás" is, mert negyven diák próbálkozásait korlátlan ideig megőrzi memóriájában. A tanár bármikor végignézheti diákjai fejlődését, és ez az információ segíti őt: például az egyéni foglalkozás hatékony lesz.

— Mik a további terveik?

— Nyelvoktató rendszerünket kiterjesztjük IBM PC kompatibilis gépekre, és más nyelvek anyagával is kiegészítjük. Mint mondottam, a program megjegyzi a felhasználók próbálkozásait. Ez a rész nem törölhető, mégpedig szándékosan nem. Bizonyos idő elteltével a felhasználóknak hajlékonylemezeket küldünk, kérve őket, hogy rátöltve a program előírt részét, küldjék vissza számunkra. Az így előálló információk, statisztikák megmutatják, hogy mely területeken kell fejlesztenünk, és mi a továbblépés útja.

(Ezt a programot még én is megvásárolnám, ha lenne otthon gépem . . .)

PINKE GYÖRGY

Színvonalas előadás

RENUMBER

COMMODORE

A RENUMBER a programok átsorszámozását jelenti. A sorozatban eddig közölt eljárások: a programtitkosítás, az overlay, append, chain és a merge mind olyan programokkal tudtak csak dolgozni, amelyekben az egyes programrészek sorszámai a feladatnak megfelelő sorrendben követték egymást. A C64 alap operációs rendszere nem végzi el a programok átsorszámozását. A BASIC-kiegészítések, a HELP vagy a Simon's BASIC ezt megteszik ugyan, de mégis hasznos lehet az alábbi program, amely ehhez a feladathoz még további szolgáltatást nyújt. A többletszolgáltatás az, hogy a programon belül önálló logikai részeket különíthetünk el, és ezeket az önálló részeket a sorszámozásban is jól különválasztva kezeljük: úgy, hogy a különálló programrészek sorszámai is különálló egységeket képezzenek. Az így kialakított program gyorsan áttekinthető.

A RENUMBER és a korábban közölt overlay, append, chain, merge eljárások együttesen a programírásnak egy haladóbb módszerét képviselik. A különálló részfeladatokat önálló szubrutinokként írjuk meg. A gyakran használt szubrutinokból szubrutinkönyvtárat képezünk, majd a programjainkat ezekből a szubrutinokból mint építőkockákból rakjuk össze.

1. lista

```
100  REM    PROGRAM ATSORSZAMOZAS
110  REM   < RENUMBER  PRG. FILE >
120  REM  -- ----------------------
130  PRINT "CLR]CD]CD]"; SPC( 8);"
     CRVS]                        CRVO]"
140  PRINT  SPC( 8);"CRVS]  RENUMBER  PR
     G.FILES  CRVO]"
150  PRINT  SPC( 8);"CRVS]
           CRVO]"
160  PRINT  SPC( 8);"CRVS] PROGAM  ATSOR
     SZAMOZAS CRVO]"
170  PRINT  SPC( 8);"CRVS]
           CRVO]": PRINT "CCD]CCD]
     CCD]CCD]CCD]CCD]"
180  '
190  INPUT " ATSZAMOZANDO PRG.NEVE ";NV$
200  N$ = "IDEIGLENES":N$ =  CHR$(0)
210  INPUT " KEZDO SORSZAM          10
     CCL]CCL]CCL]CCL]";KS
220  INPUT " SORTAVOLSAG            10
     CCL]CCL]CCL]CCL]";SP
230  INPUT " BLOKK TAVOLSAG 10TN N   3
     CCL]CCL]CCL]";BL
240  IF BL < 1 OR BL > 4 THEN  PRINT "
     CCU]": GOTO 230
250  LMAX = 5000: DIM LN(<LMAX,1>
260  OPEN 15,8,15: PRINT "CHOME]":
      GOSUB 3150
270  PRINT# 15,"S0:IDEIGLENES": GOSUB 31
     50
280  GOSUB 1030: GOSUB 2030
290  PRINT# 15,"S0:" + NV$: GOSUB 3150
300  PRINT# 15,"R0:" + NV$ + "=0:IDEIGLE
     NES": GOSUB 3150
310  GOSUB 3030: GOTO 3190
320  '
```

```
1000  REM#   ELSO MENET
1010  REM  ----------------
1020  '
1030  PTR = 0:NL = KS: PRINT "CHOME]CCD]
      CCD]CCD]CCD]CCD]CCD]CCD]CCD]   REG
      I SORSZAM:      UJ SORSZAM:"
1040  OPEN 3,8,3,NV$: GOSUB 3150
1050  GET#3,T$: GET#3,T$
1060  GET#3,T$: GET#3,T$
1070  IF T$ = "" THEN  CLOSE 3: RETURN
1080  GET#3,T#:T =  ASC<T$ +  CHR$(0))
1090  GET#3,T#:T =  ASC<T$ +  CHR$(0))
       * 256 + T
1100  PRINT "CHOME]CCD]CCD]CCD]CCD]CCD]
      CCD]CCD]CCD]CCD]CCD]     ";T
1110  GET#3,T$
1120  IF T$ =  CHR$(143) THEN  GET#3,T$:
       IF T$ = "#" THEN NL =  INT(1 + NL
      / 10 ↑ BL) * 10 ↑ BL
1130  IF T$ <  > "" THEN 1110
1140  LN(<PTR,1> = NL
1150  LN(<PTR,0> = T
1160  NL = NL + SP
1170  IF NL > 63999 THEN T$ = "TUL NAGY
      SORTAVOLSAG": GOTO 3240
1180  PTR = PTR + 1
1190  IF PTR <  = LMAX THEN 1060
1200  T$ = "TUL SOK ATSZAMOZANDO SOR":
       GOTO 3240
1210  RETURN
1220  '
```

2. lista

A négy listában közölt RENUMBER eljárás nem a tárban levő programot sorszámozza át, hanem egy, a lemezen lévő, kiválasztott programfájl átsorszámozását végzi. Az átsorszámozás előtt az elkülönítendő részeket a programban meg kell jelölni. Ezt egy REMARK utasítás után közvetlenül írt kettős kereszttel (SHIFT 3) végezzük el. Például: 130 REM # EZ EGY KÜLÖN EGYSÉG

Az így előkészített programot lemezen tároljuk, majd betöltjük és elindítjuk a RENUMBER programot. A program kérdéseire az alábbi paramétereket kell megadni:

— az átsorszámozandó program neve,
— a kezdő sorszám értéke,
— a sorszámok közötti lépésköz (sortávolság),
— a különálló blokkok közötti lépésköz a 10-es hatványaként (blokktávolság).

Az itt közölt program saját magát sorszámozta át ezekkel a paraméterekkel:

— kezdő sorszám: 100
— sortávolság: 10
— blokktávolság: 3 (azaz 1000)

A program működése

A program megértéséhez ismerni kell a tárolás alapelveit. Mivel a program a lemezen tárolt programot sorszámozza át, itt most a lemezen történő tárolást vizsgáljuk.

A programfájl első két bájtja a program betöltési címét adja. Ezután következnek az egyes utasításrok azonos felépítéssel.

Az átsorszámozásnál a sor eleji sorszámokon kívül az ugróutasítások után levő sorszámokat is megfelelően átszámítva kell felírni. Ezért az átsorszámozás két menetben halad.

Az első menetben a program a lemezről folyamatosan olvassa az egymást követő bájtokat. A mutatókat figyelmen kívül hagyja. A sorszámokat egy kétoszlopos táblázatba helyezi el.

3. lista

```
2000  REM# MASODIK MENET
2010  REM  ----------------
2020  '
2030  OPEN 3,8,3,NV$: GOSUB 3150: PRINT
      "CHOME]CCD]CCD]CCD]CCD]CCD]CCD]
      CCD]CCD]CCD]CCD]CCD]                "
2040  OPEN 4,8,4,N$ + ",P,W": GOSUB 315
      0
2050  GET#3,T$: PRINT# 4, LEFT$(T$ + N$,
      1);: GET#3,T$: PRINT# 4, LEFT$(T$
       + N$,1);
2060  FOR J = 0 TO PTR - 1:ID = 0
2070  GET#3,T$: PRINT# 4, LEFT$(T$ + N$,
      1);: GET#3,T$: PRINT# 4, LEFT$(T$
       + N$,1);
2080  GET#3,T$: GET#3,T$
2090  PRINT "CHOME]CCD]CCD]CCD]CCD]CCD]
      CCD]CCD]CCD]CCD]CCD]     "; STR$(LN
      %(J,0));"              "; STR$(LN(<
      J,1>);
2100  T =  ABS(LN(<J,1>)
2110  GOSUB 3310: PRINT# 4, CHR$(LO);
      CHR$(HI);
2120  GET#3,T$
2130  IF T$ =  CHR$(34)  THEN  ID =
       NOT  ID
2140  PRINT# 4, LEFT$(T$ + N$,1);
2150  IF <T$ =  CHR$(137) OR T$ =  CHR$(
      141) OR T$ =  CHR$(167) OR T$ =
      CHR$(138)) AND ID = 0 THEN 2200
2160  IF T$ <  > "" THEN 2120
2170  NEXT J
2180  PRINT# 4,N$,N$: CLOSE 3: CLOSE 4:
       RETURN
2190  '
2200  GET#3,T$: IF T$ = " " THEN
       PRINT# 4,T$;: GOTO 2200
2210  IF T$ < "0" OR T$ > "9" THEN 2140
2220  T1$ = T$
2230  GET#3,T$
2240  IF T$ = " " THEN 2230
2250  IF T$ >  = "0" AND T$ <  = "9"
       THEN T1$ = T1$ + T$: GOTO 2230
2260  T =  VAL(T1$):T1 =  - 1:I = 0
2270  IF LN(<I,0> = T THEN T1 = I  GOTO
      2300  ELSE
2280  IF I < PTR THEN I = I + 1: GOTO 22
      70
2290  IF T1 < 0 THEN T1$ = "????":LN(<J,
      1> =  - ABS(LN(<J,1>)): GOTO 2310
2300  T1$ =  MID$( STR$(LN(<T1,1>),2>
2310  PRINT# 4,T1$;
2320  IF T$ = "," THEN  PRINT# 4,T$;:
       GOTO 2200
2330  PRINT# 4, LEFT$(T$ + N$,1);: GOTO
      2160
2340  '
```

```
3000  REM# HIBAKEZELES
3010  REM  ----------------
3020  .
3030  PRINT "{CLR}{CD}{CD}HIBAS SORSZAMO
      K A KOVETKEZO SOROKBAN:"
3040  HS = 0
3050  FOR I = 0 TO PTR - 1
3060  IF LN%(I,1) < 0 THEN  PRINT  - LN%
      (I,1):HS = HS + 1
3070  NEXT
3080  IF HS = 0 THEN  PRINT  SPC( 17);"N
      INCS"
3090  PRINT "{CD}{CD}{CD}"
3100  RETURN
3110  .
3120  REM   DISC HIBA
3130  REM  -----------
3140  .
3150  INPUT# 15,HI,HI$
3160  IF HI < 20 THEN  RETURN
3170  PRINT "{CLR}{CD}{CD}{CD}{CD}{CD}{CD
      }{CD}{CD}{CD}     DISC HIBA"
3180  PRINT " ";HI;HI$
3190  CLOSE 3: CLOSE 4: CLOSE 15: END
3200  .
3210  REM   PARAMETER HIBA
3220  REM  ----------------
3230  .
3240  PRINT "{CLR}{CD}{CD}{CD}{CD}{CD}   PAR
      AMETER HIBA"
3250  PRINT "  ";T$
3260  GOTO 3190
3270  .
3280  REM   L/H BYTE
3290  REM  ----------
3300  .
3310  HI =  INT(T / 256):LO = T - HI * 2
      56: RETURN
```

4. lista

Minden sor egymás mellett tartalmazza a régi és a hozzá tartozó új sorszámot.

A második menetben ismét végigolvassuk a teljes programfájlt, és azt néhány változtatással egy új, „IDEIGLENES" elnevezésű programfájlba visszaírjuk. A változtatások a következők.

A régi sorszám helyett az annak megfelelő új sorszámot másoljuk át.

A GOTO, GOSUB, RUN, THEN után álló sorszámok helyett a táblázatból kikeresett új sorszámokat írjuk fel az új programba. Ha nem létező sorszámra hivatkoznak ezek az utasítások (UNDEFINED LINE), akkor a programba a sorszám helyére ???? kerül. A program végén ezeket az utasításokat összegyűjtve a képernyőn felsoroljuk. A sorszámozás befejezése után az eredeti programfájlt töröljük a lemezről, majd a most készült IDEIGLENES elnevezésű programfájlt átkereszteljük az eredeti program nevére.

Fontosabb változók a programban

NV$ — a programfájl neve
NW$ — az ideiglenes program neve
N$ — a bájtot tartalmazó konstans
KS — a kezdő sorszám
SP — sortávolság
BL — blokktávolság
LM — a maximális sorok számát adó konstans
NL — új sorszám
HS — hibás sorszámokat tartalmazó sorok száma
TS — az olvasott bájt vagy kiírandó szöveg
T — az olvasott bájt ASCII kódja
ID — idézőjeljelző
LO — kétbájtos szám alsó bájtja (LOW)
HI — kétbájtos szám felső bájtja (HIGH)
LN%(LM,1) — a sorszámokat tartalmazó táblázat

A program vázlatos ismertetése

Főprogram (1. lista)

100—260 paraméterek megadása
270 az esetleges IDEIGLENES nevű program törlése
280 az átsorszámozás két menetének hívása
290 az eredeti program törlése
300 az IDEIGLENES program átkeresztelése
310 hibás sorszámok kiírása

Első menet (2. lista)

1030—1040 kezdeti beállítások
1040—1070 a mutatók átlépése
1080—1100 sorszám beolvasása, átkódolása, kiírása
1110 a következő bájt olvasása
1120 REM # vizsgálata
1130 az utasítássor vége
1140—1150 sorszám elhelyezése a táblázatban
1160—1210 új sorszám kiszámítása és ellenőrzése

Második menet (3. lista)

2030—2040 kezdeti beállítások
2050—2070 mutatók olvasása, írása
2080—2090 sorszám olvasása, kijelzése
2100—2110 új sorszám kikeresése a táblázatból és felírása
2120—2170 a következő bájt olvasása, vizsgálata és felírása
2180 programvégjel felírása
2200—2340 GOTO stb. utáni sorszámok kicserélése

Segédrutinok (4. lista)

3000—3100 hibás sorszámot tartalmazó sorok kijelzése
3120—3190 lemezhiba-vizsgálat és -kezelés
3210—3270 hibás paraméterezés kezelése
3280—3310 L/H bájt kiszámítása

ZSOM BÉLA

Rajzprogramunkról nagyon hamar bebizonyosodik, hogy számos hiányossága van. Talán a legelső észrevétel, hogy teknősünk mindig rajzol, vagyis képtelen arra, hogy az egyik pontból a másikba úgy jusson el, hogy ne húzzon vonalat. A LOGO parancsai között van egy pár, ami ennek a problémának a kiküszöbölésére szolgál: a PU (tollat fel), PD (tollat le) parancsok (lásd az *ábrát*). Beépítésük programunkba viszonylag egyszerű, hiszen ezek a parancsok nem kérnek paramétert, vagyis nem áll utánuk szám. E parancsok beépítéséhez két dolog szükséges: a parancsok felsorolásába be kell kerülniük, továbbá a parancsok végrehajtását végző programrészbe is. Az első igen egyszerű, hiszen csak annyi a teendőnk, hogy a parancsokat felsoroló részt ki kell egészítenünk azzal a szöveggel, hogy „PU-toll fel", illetve „PD-toll le". A megvalósító részbe sem nehéz a beépítés, ha megjegyezzük, hogy oda kell betennünk, ahol még nem dőlt el, hogy mit is tegyen a teknős. A programrész konkrét megvalósítása akkor lehetséges, ha tudjuk, hogy gépünk milyen módon rajzol, illetve nem rajzol egyenes szakaszt. Sok gépnél, amelyeknél egyenest rajzoló utasítás vagy pontrajzoló utasítás létezik, a programrész megvalósítása egyszerűen ez utasítások paraméterezése. A pontrajzoló utasítással az egyenesrajzolót helyettesítjük úgy, hogy pontsorozattal jelképezzük az egyenest. Az én gépemnél a paraméterezés még egyszerűbb, csak a RAJZ$ nevű változóba kell betennünk vagy onnan kivennünk egy karaktert. Ezt a programlista 130-as és 140-es sora mutatja (lásd a *listát*). Azt ugyanis, hogy ha a RAJZ$ nevű változó hossza 10 karakter, akkor idáig rajzolt (mert a RAJZ$ nem tartalmazta a B betűt, ami a rajzolást megtiltó jel), és vagy hozzá kell írni a B betűt, ha nem akarunk rajzolni, vagy változatlanul ki kell hagyni. Ha a RAJZ$ 11 karakter hosszúságnyi volt és rajzolni akarunk, el kell vennünk belőle az első karaktert, ha pedig nem, akkor maradjon változatlanul.

Más gépeknél egy változó értékét kell 0-ra vagy 1-re állítanunk.

Bár elkészítettünk valamilyen szép rajzot, szomorúan kell megállapítanunk, hogy még nem gondoskodtunk a megörökíthetőségéről. A képernyő fényképezése egyrészt nehézkes, másrészt a kép állandó vibrálása és a fényerőproblémák miatt igazán jó minőségű képet sohasem szolgáltat. Könnyebb dolga van, akinek van nyomtatója, mert a „nyomtasd" utasítással ki tudja egészíteni a programot. E programrész elkészítése előtt azonban meg kell ismerkednünk azzal, hogyan is tudunk egyáltalán képet nyomtatni.

Teknősbéka-grafika II.

A jelenleg kapható nyomtatók mindegyike képes ún. grafikus üzemmódban dolgozni. Ez azt jelenti, hogy pontonként lehet megadni, hogy mit nyomtasson. Tételezzük fel, hogy a számítógép által a képernyőre és a nyomtatóra küldött képelemek egymáshoz képesti elhelyezkedése és a képelemek száma is azonos. A kép mégsem lesz méretarányos (mert a képernyőn a képelemek vízszintes irányú kiterjedése kisebb, mint a függőlegesek, míg a nyomtatóban ez a két kiterjedés általában azonos). Ráadásul nagyon kis méretű lesz maga a nyomtatott kép (mert amíg a képernyőn a legtöbb gépnél a vízszintes irányú pontok száma 256 vagy 320 és csak különleges esetekben ennél nagyobb, addig a nyomtatóknál 960 vagy 1920). Ezeket is figyelembe kell vennünk a nyomtató rész elkészítésénél.

1. lista

```
10 REM RAJZOLO II
20 PRINT"PARANCSOK:":PRINT"FD-ELOERE":PR
INT"BK-HAATRA":PRINT"LT-BALRA":PRINT"RT-
JOBBRA":PRINTMINDEGYIK UTAAN EGY SZAAM A
ALL.":?"PU-TOLL FEL":?"PD-TOLL LE":?"U-N
YOMTATAAS":?"A PARANCSOK SOROZATBA KAPCS
OLHATOOK."
30 PRINT"AZ ELVAALASZTOJEL A PONTOSVESSZ
OE."
40 CLEAR2000:X0=128:Y0=96:PI=355/113:ALF
A=PI/2:FI=ALFA:X=X0:Y=Y0:PMODE4,1:PCLS
50 B$=INKEY$:IFB$="" THEN50 ELSE INPUTA1
$:A=LEN(A1$):RAJZ$="M "+X$+","+Y$
60 IFA1$="U" THEN GOSUB270:STOP
70 FORI=1TO A:J=0
80 J=J+1:IF J=A-I+1 THEN100
90 J$=MID$(A1$,I-1+J,1):IF J$<>";" THEN8
0
100 A$=MID$(A1$,I,J):I=I+J-1
110 IF J$=";" THEN A$=LEFT$(A$,LEN(A$)-1
)
120 IF FI>2*PI THEN FI=FI-(2*PI)
130 IF (A$="PU")AND(LEN(RAJZ$)=10) THEN
RAJZ$="B"+RAJZ$:GOTO260
140 IF (A$="PD")AND(LEN(RAJZ$)=11) THEN
RAJZ$=RIGHT$(RAJZ$,LEN(RAJZ$)-1):GOTO260
150 C$=LEFT$(A$,2):D$=MID$(A$,3,LEN(A$)-
2):D=VAL(D$):SCREEN1,1:IF (C$="FD")OR(C$
="BK") THEN220
160 IF C$="LT" THEN200
170 IF C$="RT" THEN190
180 GOTO260
190 D=360-D
200 FI=FI+(D*PI/180)
210 GOTO260
220 IF C$="BK" THEN D=-D
230 X=X+(D*COS(FI)):Y=Y-(D*SIN(FI))
240 IF(X>-1)AND(X<256)AND(Y>-1)AND(Y<192
) THEN DRAW RAJZ$:GOTO290
250 IFX<0THEN X=0
260 IFX>255THEN X=255
270 IFY<0THEN Y=0
280 IFY>191THEN Y=191
290 NEXTI:GOTO50
300 PRINT#-2,CHR$(13);CHR$(24);CHR$(27);
"0";CHR$(27);"T08";:AA=1536:FORII=0TO191
STEP4:DD=32*II+AA:FORJJ=0TO31:Y1(0)=PEEK
(JJ+DD):Y2(0)=PEEK(JJ+32+DD):Y3(0)=PEEK(
JJ+64+DD):Y4(0)=PEEK(JJ+96+DD):C1=256:FO
RKK=1TO8:C1=C1/2:B1=C1*INT(Y1(KK-1)/C1)
310 Y1(KK)=Y1(KK-1)-B1:B2=C1*INT(Y2(KK-1
)/C1):Y2(KK)=Y2(KK-1)-B2:B3=C1*INT(Y3(KK
-1)/C1):Y3(KK)=Y3(KK-1)-B3:B4=C1*INT(Y4(
KK-1)/C1):Y4(KK)=Y4(KK-1)-B4:PRINT#-2,CH
R$(27);"W0008";:IFB1<>0THEN EE=15:GOTO32
0 ELSE EE=0:GOTO320
320 IFB2<>0THEN PRINT#-2,CHR$(EE+128);:G
OTO330 ELSE PRINT#-2,CHR$(EE);:GOTO330
330 IFB3<>0THEN EE=15:GOTO340 ELSE EE=0:
GOTO340
340 IFB4<>0THEN PRINT#-2,CHR$(EE+128);:G
OTO350 ELSE PRINT#-2,CHR$(EE);:GOTO350
350 NEXT KK:NEXT JJ:PRINT#-2,CHR$(13);:N
EXT II:RETURN
```

A nyomtatót ebben a programban ún. karakter üzemmódban használjuk, amelyben a számítógép ASCII kódok formájában küldi a nyomtatónak a leírnivalókat. Egy-egy ilyen kódnak megfelelő karakter előállítására mind a képernyőn, mind a nyomtatón egy ún. karaktermátrix áll rendelkezésre. Ez a képernyőn, illetve a papíron a kiíratás eszköz által egy karakter számára elfoglalt hely pontokban megadva. Itt is eltérések vannak a képernyő és a nyomtató között. (Ez alól azok a gép—nyomtató párok képeznek kivételt, amelyek egy adott számítógéphez tervezett nyomtatóból és magából a gépből állnak, mint például a C64 és MPS—801.) A karaktermátrix mérete mind a képernyőn, mind a nyomtatón igen különböző. Függ a számítógéptől és a nyomtatótól. A legtöbb olcsó mikroszámítógép a képernyőjére 8×8 pontból álló mátrixot küld. (Az én képernyőm 8×16-os.) A legelterjedtebb nyomtatók karaktermátrixai a 8×8 ponttól felfelé, a 18×24-es dimenziókig széles skálán belül változnak. E nyomtatók mindegyikéhez más és más átszámításra van szükség. (Az én nyomtatóm NECPR 103A, olcsó ára ellenére 12×18 pontból állít elő egy karaktert.) A nyomtatók működésük közben igazában karakterenként állítják elő a rajzokat is, csak olyan karakterekként, amelyeket pontokból (képelemekből) magunk állítunk össze. Saját nyomtatómnál vízszintes irányban egy karakternek grafikus üzemmódban 1—1920 pontnyi szélesség is megadható.

Ez lehetőséget ad rá, hogy a mérettorzítást kiküszöböljük, egyúttal nagyítsunk azáltal, hogy a képernyőelemek számát ebben a tartományban transzformáljuk. Az ún. grafikus képterület (a memóriának az a része, ahol a képek tárolódnak) pontjait a kép bal felső sarkától soronként kell a nyomtatóra küldeni, mindaddig, amíg el nem érjük a jobb alsó sarkot. Ha nem akarunk pici képet kapni, célszerű minden egyes pontot egy 8×8 méretű mátrixszá nagyítani, vagyis a képünk méretét mindkét koordináta irányába megnyolcszorozzuk. (Ez az eljárás a teljes grafikus képterület kinyomtatására azonban csak akkor tanácsos, ha a nyomtatónk vízszintes irányban legalább nyolcszor több pont kirajzolására

```
PARANCSOK:
FD-ELOERE
BK-HAATRA
LT-BALRA
RT-JOBBRA
MINDEGYIK UTAAN EGY SZAAM AALL.
PU-TOLL FEL
PD-TOLL LE
U-NYOMTATAAS
A PARANCSOK SOROZATBA KAPCSOLHATOOK.
AZ ELVAALASZTOJEL A PONTOSVESSZOE.
```

1. ábra

képes, mint a képernyő maximális vízszintes irányú képelemeinek száma. Egyébként ez a kényelmes módszer nem teszi lehetővé a teljes képterület megjelenítését. Én is 256-ról 240 pontra korlátoztam a vízszintes koordinátatartományt.)

A tényleges nyomtatáshoz a nyomtatón a kívánt üzemmódot kell kiválasztani, vagyis a megfelelő felbontású grafikus változatot, és be kell állítanunk azt, hogy soremeléskor hány pontot emeljen. (Ha ezek nem állíthatók valamely nyomtatón, akkor különleges megoldásokat kell alkalmazni, amihez a segítséget ezúton is felajánlom!)

E paraméterértékek beállítása után elő kell vennünk a grafikus képterületről az egyes pontoknak megfelelő értékeket. Újabb feladatok várnak ránk. A számítógépek tárjainak tartalmához 8 bites gépeknél a felhasználó bájtformában tud hozzáférni. Nekünk azonban nem erre, hanem minden egyes bitértékre külön-külön van szükségünk. Szét kell tehát bontanunk minden egyes bájtot bitekre, és ez küldendő majd a nyomtatónkra. Egy bájt felbontása ilyen módon abból áll, hogy helyi értékek szerint szétválogatjuk. Egy bájt egy 256-os rendszerbeli szám egy helyi értékét képviseli, ezt kell tehát kettes számrendszerbeli számmá átalakítani. Az átalakítás csökkenő sorrendben 128-cal, 64-gyel stb. osztással halad. Először tehát el kell osztani 128-cal számértékünket (ez lesz a legmagasabb helyi érték kettes számrendszerbeli számunkban), és meg kell vizsgálnunk, hogy a kapott érték nagyobb-e 1-nél: ha igen, első számjegyünk 1, ha nem, 0. Ezután az így kapott első helyi értéket megszorozzuk 128-cal, és az eredményt levonjuk számunkból. A kivonás eredményével fogunk tovább dolgozni. Az előzőekben ismertetetteket megismételjük 64-gyel, 32-vel stb., összesen nyolcszor. Az így előállt bitsorozat már alkalmas a nyomtatóra küldésre.

Nyomtatónk tulajdonságainak ismeretében jól tudjuk szabályozni a nagyítás, torzítás kiküszöbölését. Saját nyomtatómnál négy egymás alatti pontot vízszintes irányban megnyolcszorozva állítottam elő egy-egy karaktert. Mivel gépemnél a képernyőn

vízszintes irányban 256 pont van, így az egymás alatt levő pontok tárcímeltérése 32.

A nyomtató részt programunk 60., 300—350. sorai tartalmazzák. A 60. sorban megvizsgáljuk, hogy volt-e nyomtass (ü) parancs: ha igen, menj a nyomtató szubrutinra, majd állj le. A nyomtató részben először sort emelünk, töröljük a nyomtató belső memóriáját, bekapcsoljuk a grafikus üzemmódot, beállítjuk a soremelés mértékét. Ezzel a nyomtatót előkészítettük. Megcímezzük a grafikus képterület első helyét (AA).

Az II ciklussal soronként, a JJ ciklussal bájtonként végiglépkedünk a grafikus képernyőterületen. A négy elővett bájt az Y nevű változóvektorok 0. elemének ad értéket. Ezekből számítjuk a bitértékeket. Kettőnek az osztáshoz szükséges hatványértékét a C1 nevű változó tartalmazza. Ez először 256.

A bájtszétbontást a KK ciklus végzi. Először C1 értékét felezzük. A bájt megmaradt darabjából az osztás, egészérték-képzés, szorzás után megkapjuk azt a számot (B), amely a bitértékre jellemző (ha ez zérus, akkor a bitérték is), és ezt a bájtmaradékból levonva előállítjuk az újabb bájtmaradékot.

A nyomtatón beállítjuk a vízszintes irányú nyolcszorozást. B-k értékéből a négy függőleges irányú pontot (EE) állítjuk be. Az utolsó sor a ciklusok lezárása, soremelés, sor elejére állás.

Dr. SIMONYI ENDRE

Közületek, figyelem!

Mikroszámítógépet akarnak vásárolni? Tájékozódjanak a naprakész piaci helyzetről!

Díjtalan ismertető!

MESZ Számítástechnika
1368 Budapest, Pf. 193.

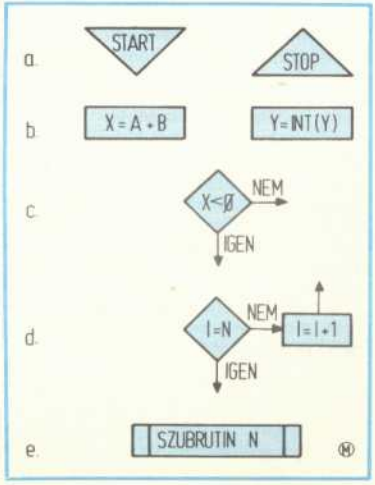
Fo

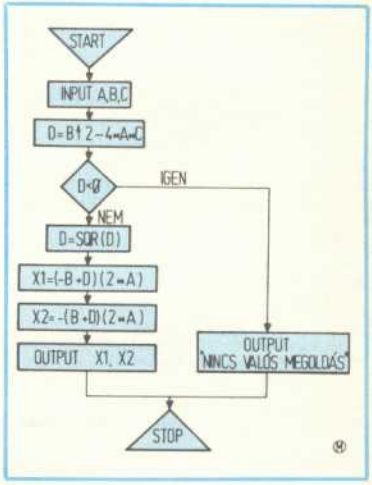
Sokat bosszankodunk, ha programunk nem viselkedik megfelelően, s még jobban, ha a hibát sem tudjuk felfedezni. Programunkat futtathatjuk lépésenként, beszúrhatunk PRINT utasítást, mely tájékoztat a változók értékéről, de gyakran csapdába esünk. Nem könnyű a saját magunk logikájából kitörni! Ilyenkor szoktuk barátunkat, ismerősünket megkérni: „Ugyan nézd már át, mert nem találom a hibát!" Velem is előfordult, hogy végül a gépet is meggyanúsítottam, hátha ő téved. Erre azonban még nem volt példa . . .

A folyamatábra (blokkdiagram) nem old meg minden programozással kapcsolatos kérdést, de hasznos segítőtársunk lehet. Elsősorban a programozás tanulásához elengedhetetlen, de számos előnye miatt sokan ragaszkodnak hozzá később is.

A folyamatábra elkészítése először is megkönnyíti a programok szerkesztését. Néha még a leggyakorlottabb programozó is csinál folyamatábrát programja bonyolultabb részeiről, mivel így a működése, logikája sokkal áttekinthetőb. Nagy előnye a módszernek, hogy nem kötődik konkrét programozási nyelvhez; ha mi közlünk néha így egy-egy algoritmust, akkor a különböző gépekkel rendelkező olvasóink is viszonylag könnyen elkészíthetik saját programjukat, és nem kész „recept" alapján billentyűzik be a közölt megoldást.

1. ábra





2. ábra

Miért segít a folyamatábra a hibakeresésben? A programsorok egymás után következnek, s végül maga a megalkotójuk is elveszti az algoritmus fonalát. A program elágazásait, a vezérlésátadásokat a folyamatábrán simán követhetjük.

A csúcsával lefelé fordított háromszög a program kezdetét, a felfelé fordított háromszög a végét jelöli. A háromszögekbe beírjuk a START és STOP szavakat. *(1. ábra, a.)*

Téglalapba írjuk a közvetlenül végrehajtható utasításokat *(1. ábra, b.)*. Példánkban X legyen egyenlő A és B változó összegével. Téglalapba írjuk a függvényutasításokat is: például Y legyen egyenlő az egész részével. Az adatok beolvasása és kiíratása közvetlenül végrehajtható utasítások, így azokat is téglalapban jelöljük, az INPUT, OUTPUT szavakkal. Ezt általánosságban értjük, az adott programnyelvben lehet READ, WRITE, PRINT stb. utasítás.

A feltételes elágaztatást csúcsára állított rombusz jelöli. A rombuszból kivezető nyíllal jelzett vonalak mutatják a program viszszatérését. Ha a feltétel teljesül, akkor a program az IGEN ágon, ha nem, akkor a NEM ágon folytatódik. A program vezérlé-

matábrák

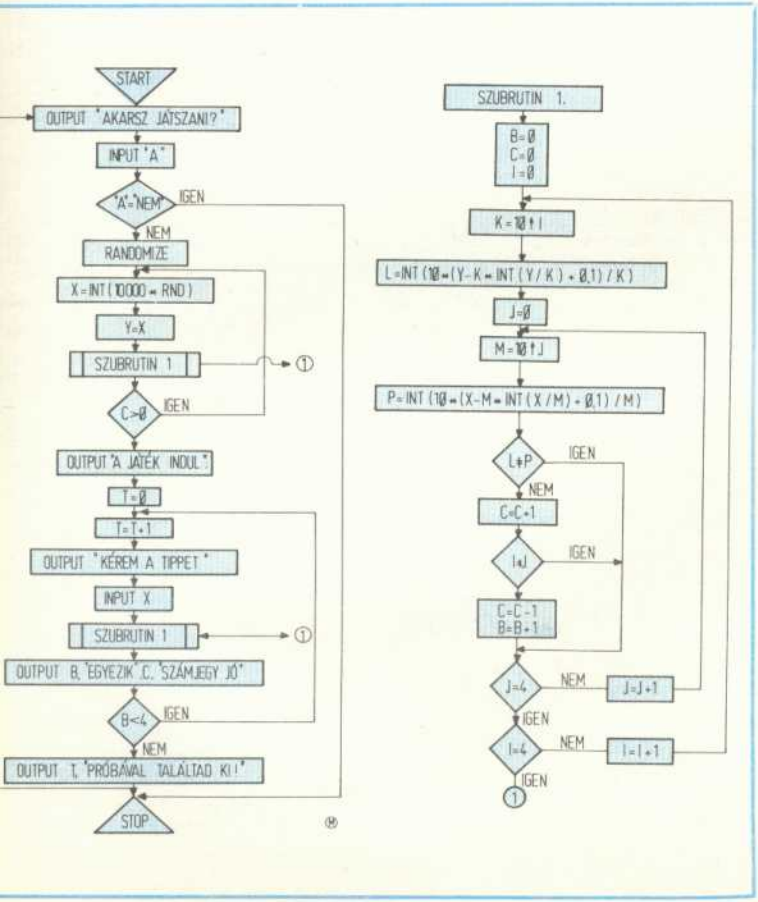
se a nyillal jelzett vonalak csatlakozó pontjára kerül át *(1. ábra, c.)*.

Sok problémát okoz, hogy a ciklusváltozó kezdőértékeit a különböző programozási nyelvek nem egyformán kezelik. Ezért a folyamatábrában a ciklus megszervezésére a feltételes elágaztatást használjuk. Ez könnyen felismerhető, mivel külön jelöljük a ciklusváltozó értékét. Figyeljünk arra, hogy a ciklusváltozó kezdőértékét meg kell adni *(1. ábra, d.)*.

A programozásban gyakori a szubrutinok alkalmazása. A folyamatábrában a szubrutin hívására is alkalmazunk egy jelölést *(1. ábra, e.)*. A szubrutin után feltüntetjük a hívását jelző számot. A szubrutinoknak külön-külön folyamatábrát készítünk, melyek önálló programkészleteket alkotnak.

3. ábra

4. ábra




```
10 PRINT "AKARSZ JATSZANI ?"
100 INPUT A$
105 IF A$="NEM" THEN GOTO 1200
106 RANDOMIZE
110 LET X = INT(10000*RND)
120 LET Y = X
130 GOSUB 1000
140 IF C>0 THEN GOTO 110
150 REM VAN-E AZONOS SZAMJEGY
160 PRINT "A JATEK INDUL"
170 LET T = 0
180 LET T = T+1
190 PRINT "KEREM A TIPPET"
200 INPUT X
210 GOSUB 1000
220 PRINT B;"EGYEZIK",C;"JO SZAMJEGY"
230 IF B<4 THEN GOTO 180
240 PRINT T;"PROBAVAL TALALTAD KI"
250 PRINT
260 GOTO 10
1000 REM SUBRUTIN
1010 LET B = 0
1020 LET C = 0
1030 FOR I = 1 TO 4
1040 LET K = 10^I
1050 LET L = INT(10*(Y-K*INT(Y/K)+.1)/K)
1060 FOR J = 1 TO 4
1070 LET M = 10^J
1080 LET P = INT(10*(X-M*INT(X/M)+.1)/M)
1090 IF L<>P THEN GOTO 1140
1100 LET C = C+1
1110 IF I<>J THEN GOTO 1140
1120 LET C = C-1
1130 LET B = B+1
1140 NEXT J
1150 NEXT I
1160 RETURN
1200 END
```

A *2. ábra* egyszerű programozási feladat folyamatábráját mutatja be, a másodfokú egyenlet általános megoldását. A másodfokú egyenlet általános alakja és megoldása:

$$ax^2+bx+c=0$$

$$x_{1,2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

Az a, b, c meghatározott érték. A folyamatábra szerinti algoritmus beolvassa a, b, c értékét, és kiszámítja a gyök alatti értéket.

Megvizsgálja továbbá, hogy az pozitív-e. Ha nem, akkor kiírja: NINCS VALÓS MEGOLDÁS, egyéb esetben kiszámolja x értékeit.

Jó tanácsként megemlítjük, hogy ha a folyamatábrában egymás mellett van két végrehajtandó utasítás, akkor ez a programban majdnem minden esetben egy közvetlen vezérlésátadó utasításnak felel meg! A megoldást a 2. ábra mutatja.

Végül egy logikai játék folyamatábráján egyúttal egy általános BASIC programot is ismertetünk. A játék programja Donald Alcock Ismerd meg a BASIC nyelvet c. könyvében található *(3. ábra* és *4. ábra)*.

A program ismertetése előtt ejtsünk néhány szót a véletlenszámokról. A véletlenszámok — nem matematikai definícióval — egymástól különböző, független számok sorozataként képződnek. Sokan azt hiszik, hogy ha egymás után gondolnak számokra, azok véletlenszámok, de könnyen bebizonyítható, hogy nem, mivel akaratlanul is valamilyen logikát követnek. Következő cikkünkben közlünk egy ún. számítógépre írt bűvészprogramot, amely kis hibával bármely gondolt számot „kitalál". A megoldást nem akarjuk előre közölni, de a program egyértelműen arra épül, hogy a találomra mondott számok között is van összefüggés, ha nem is gondolunk rá.

A játék programja véletlenszámot állít elő, melyet a játékosnak ki kell találnia. A nálunk „mastermind" néven ismert játék lényege, hogy négy számjegyre tippelünk. Egy számban a számjegyek között azonosak nem lehetnek, és két tipp sem lehet megegyező. A program 110-es sora előállít egy véletlenszámot. A 130-as sor szubrutinja megvizsgálja, hogy a véletlenszám számjegyei között nincs-e azonos. Amennyiben azonos számjegy van, úgy új véletlenszámot állít elő. Az 1050-es sor a véletlenszámot bontja fel négy számjegyre, az 1080-as a gondolt számot, a tippet bontja fel számjegyekre.

A további utasítások megvizsgálják, hogy hány számjegy egyezik. Ha a szám és helyiértéke is egyezik, akkor a B változó értéke növekszik annyival, amennyi helyes találat van. Ha a tippelt szám helyiértéke nem egyezik, de a számjegy jó, akkor a C változó értéke növekszik. A program kiírja, hogy a játékos hány számot és hány számjegyet talált ki. Végül megtudhatjuk, hogy hány próbálkozásból találtuk ki a négyjegyű számot, s a gép megkérdezi, hogy akarunk-e tovább játszani.

A RANDOMIZE utasítás a véletlenszám előállítását mindig más kezdőértékre vezérli. Ha ez az utasítás nincs a programnyelvben, akkor az RND-re például lehet ciklust szervezni. A ciklus végét változóval kell megadni, amelyet véletlenszerűen kell a program indításakor bevinni.

PINKE GYÖRGY

Vigyáza

1987. áprilisi és májusi számainkban a 21-es és a 20-as hiba generálására láttunk példát a DOS Inside programja alapján. A 21-es hiba felírásánál az illető sávot vagy blokkot (beleértve az adatrészt és a hozzá

1. lista

```
100 REM 22-ES TÍPUSÚ HIBA GENERÁLÁSA
101 REM
105 REM INPUT PARAMÉTEREK: TRACK (SÁV)
106 REM ÉS SZEKTORSZÁM
107 REM
110 PRINT"[CLR]22-ES TÍPUSÚ HIBA"
120 PRINT"KÉREM AZ ILLETŐ BLOKK SÁV ÉS
122 PRINT"SZEKTOR SZÁMÁT (T,S)"
130 INPUT"(T,S)";T,S
140 IF T 1 OR T 35 THEN END
150 NS=20+2 (T 17)+(T 24)+(T 30)
160 IF S 0 OR S NS THEN END
170 INPUT"AZ ADATOK HELYESEK? (I/N)";Q
180 IF Q   "I" THEN END
190 OPEN15,8,15
200 PRINT 15,"IO"
210 INPUT#15,EN ,EM ,ET ,ES
220 IF EN ="00" GOTO 270
230 PRINT EN ","EM ","ET ","ES
240 CLOSE15
250 END
260 REM
270 JOB=176
280 GOSUB 440
290 IF E  1 GOTO 550
300 REM
310 JOB=128
320 GOSUB 440
330 IF E  1 AND E  4 AND E  5 THEN 550
340 PRINT#15,"M-W"CHR (71)CHR (0)CHR (1)CHR (6)
360 JOB=144
370 GOSUB 440
380 PRINT#15,"M-W"CHR (71)CHR (0)CHR (1)CHR (7)
390 IF E  1 GOTO 550
400 CLOSE15
410 PRINT"KÉSZ!"
420 END
430 REM
440 TRY=0
450 PRINT#15,"M-W"CHR (8)CHR (0)CHR (2)CHR (T)CHR (S)
460 PRINT#15,"M-W"CHR (1)CHR (0)CHR (1)CHR (JOB)
470 TRY=TRY+1
480 PRINT#15,"M-R"CHR (1)CHR (0)
490 GET#15,E
500 IF E ="" THEN E =CHR (0)
510 E=ASC(E )
520 IF TRY=500 GOTO 540
530 IF E 127 GOT 470
540 RETURN
550 CLOSE15
560 PRINT"[RVS]NEM SIKERÜLT"
570 END
```

Tolvaj!

2. lista

```
100 REM
101 REM 22-ES TÍPUSÚ HIBA ÁTMÁSOLÁSA
105 REM BEMENŐ ADATOK: SÁV- ÉS SZEKTORSÁM
106 REM
110 PRINT"[CLR] 22-ES TÍPUSÚ HIBAMÁSOLÁS"
120 PRINT"KÉREM A MASTER-LEMEZT A MEGHAJTÓBA!"
130 INPUT"SÁV- ÉS SZEKTORSZÁM (T,S)";T,S
140 IF T 1 OR T 35 THEN END
150 NS=20+2 (T 17)+(T 24)+(T 30)
160 IF S 0 OR S NS THEN END
170 INPUT"AZ ADATOK HELYESEK? (I/N)";Q
180 IF Q   "I" THEN END
190 OPEN15,8,15
200 PRINT#15,"IO"
210 INPUT#15,EN ,EM ,ET ,ES
220 IF EN ="00" GOTO 270
230 PRINT EN ","EM ","ET ","ES ","
240 CLOSE15
250 END
260 REM
270 JOB=176
280 GOSUB 550
290 REM OLVASÁS
300 JOB=128
310 GOSUB 550
320 PRINT#15,"M-R"CHR (56)CHR (0)
330 GETT#15,D
340 IF D ="" THEN D =CHR (0)
350 CLOSE15
360 PRINT"VEGYED KI A MASTERT A MEGHAJTÓBÓL!"
370 PRINT"TEDD AZ ÁTÍRANDÓ LEMEZT A MEGHAJTÓBA!"
380 PRINT"[RVS]RETURN - FOLYTATÁS ROFF
390 GET C : IF C ="" THEN 390
400 IF C   CHR (13) GOTO 390
410 PRINT"OK"
420 OPEN15,8,15
430 REM
440 JOB=176
450 GOSUB 550
460 PRINT#15,"M-W"CHR (71)CHR (0)CHR (1)D
470 REM ÍRÁS
480 JOB=144
490 GOSUB 550
500 PRINT#15,"M-W"CHR (71)CHR (0)CHR (1)CHR (7)
510 CLOSE15
520 PRINT"KÉSZ!"
530 END
540 REM
550 TRY=0
560 PRINT#15,"M-W"CHR (8)CHR (0)CHR (2)CHR (T)CHR (S)
570 PRINT#15,"M-W"CHR (1)CHR (0)CHR (1)CHR (JOB)
580 TRY=TRY+1
590 PRINT#15,"M-R"CHR (1)CHR (0)
600 GET#15,E
610 IF E ="" THEN E =CHR (0)
620 E=ASC(E )
630 IF TRY=500 GOTO 660
640 IF E 127 GOTO 580
650 RETURN
660 PRINT#15,"M-W"CHR (71)CHR (0)CHR (1)CHR (7)
670 CLOSE15
680 PRINT"[RVS]NEM SIKERÜLT ROFF "
690 END
```

tartozó blokkfejet is) a $55 értékkel egyszerűen felülírtuk, így az itt elhelyezett adatok megsemmisültek. A 21-es típusú hibához már csak a szektorszámot írtuk át: az adatok sértetlenek maradtak, de az illetéktelenek nem férnek hozzá. Ehhez hasonlóan most a 22-es hiba létrehozásához is csak egy bájtot, az adatblokk azonosítóját írjuk át.

Márciusi számunkban tárgyaltuk az adatblokk felépítését. Láttuk, hogy a szinkronjel után egy azonosító következik, amivel a rendszer megkülönbözteti a blokk fejrészét az adatrésztől. Ez az első esetben $08, a másodikban $07 (lásd az *ábrát*), amit a szoftver ír a lemezre, és értékét a $0047-es RAM-címen levő DBID (Data Block ID) nevű rendszerváltozó tárolja. A gép újraindításakor a változó mindig $07 értékkel töltődik fel, de ez bármikor átírható. Az FDC a szinkronjel utáni bájtot összehasonlítja a DBID változóval, és ha a kettő megegyezik, akkor a további bájtokat GCR kódban rögzített adatbájtoknak értelmezi: különben 22-es típusú hibaüzenettel tér vissza a C64 rendszerébe.

Annál a levédési technikánál, amellyel most ismerkedhetnek meg olvasóink, ennek a változónak van nagy szerepe.

Az adatok rögzítésénél egyes blokkok esetén ezt a változót átírjuk, így a lemezre más azonosító kerül. E blokkok csak az írásnál érvényes azonosítóval dekódolhatók. Ügyeljünk arra, hogy az új azonosító ne $07 legyen, mint a blokkfej azonosítójának kódja; ellenkező esetben az FDC nem tesz különbséget a blokkfej és az adatrész között. A DBID értékének megválasztását korlátozza az is, hogy az első számjegynek mindenképpen zérusnak kell lennie, mert különben az FDC hibásan érzékeli szinkronjel végét és az adatok elejét.

A példaprogramban *(1. lista)* a DBID értékét az eredetihez képest eggyel csökkentjük, vagyis a $07 alapállapotát $06-ra írjuk át. Önkényesen választottuk ezt, de az előbbi megkötések figyelembevételével bármely más értékre átírhatjuk, csak a 340-es sorban a CHR$(6) kifejezés argumentumát kell megváltoztatni. Az azonosító eredeti értékét a 380-as sorban állítjuk vissza — ezzel biztosítva, hogy programunk lefuttatása után ismét alapállapotban működtethessük a meghajtót, és a standard formattált lemezeket újra elolvashassuk.

Most pedig nézzük a *2. lista* programját, amely egy ún. master lemez megadott blokkjainak DBID-jét átmásolja az új levédendő lemezre. A program általában is hasonló eljárással levédett lemezt másol: a 320-as sorban olvassa a master lemez azonosítóját, és a 460-490-es sorokban átírja a másik lemezre; az 500-as sorban pedig visszaállítja a DBID azonosító eredeti ($07) értékét.

KOVÁCS P. ATTILA

Az adatok és a valóság

Amikor a külvilágot adatokra képezzük le, akkor a valós dolgok sok tulajdonságát figyelmen kívül hagyjuk, az általunk vizsgált probléma szempontjából elhanyagolhatónak tekintjük. Például egy olyan bonyolult valamit, mint egy eleven ember, a gépben egy rekorddal ábrázolunk. Ez a rekord tartalmazhatja az ember nevét, életkorát, a szeme színét, azt, hogy szereti-e a bablevest füstölt kolbásszal stb., mégse tükrözi az embert a maga komplex valóságában. Amikor egy élő embert vagy a valóság bármely más kiragadott részét adat formájában a gépben leképezzük (ábrázoljuk), akkor csak a szemünk előtt lebegő, a későbbi feldolgozás szempontjából lényeges tulajdonságait írjuk le. Nagyon fontos felismerés: a valós világot leíró adatok megválasztása mindig szorosan összefügg a feldolgozás céljával.

Környezetünkben nem tudnánk eligazodni, ha nem strukturálnánk. A tanulók az iskolában például nem homogén tömeget alkotnak, hanem különböző létszámú és fokozatú stb. osztályra tagozódnak. A kerületi tanácsoknak is osztályai, csoportjai vannak. Egy autó is néhány fő részből áll, melyek szerelvényekre, végül alkatrészekre bomlanak.

Aki csak a BASIC nyelvet ismeri, az csak igen egyszerű, strukturálatlan vagy csak igen primitív struktúrájú adatokban gondolkodhat. Ha egy mélyebben tagolt struktúrát szeretne jól *áttekinthetően* adatokra leképezni, a BASIC-kel már nehézségekbe ütközik. Könnyebb helyzetben van, aki például PASCAL-ban programoz.

Strukturálatlan adat egy egész szám, például a „3127". Igen egyszerű struktúrája van az alábbi, 4 darab egész számból álló tömbnek:

27	3162	38	995

A BASIC-ben egy ilyen tömböt így deklarálhatunk:

```
.
.
640 DIM A (4)
.
.
```

PASCAL nyelven pedig így:

```
.
.
var a: array [1..4] of integer;
.
.
```

A tömb 3. elemére BASIC-ben pl. így hivatkozhatunk:

```
.
.
950 LET B = A (3)
```

PASCAL-ban meg így:

```
.
.
b:=a [3];
.
.
```

Gondoljunk most egy gépkocsiszervizre, amikor átveszik az ügyféltől a gépkocsit. Ilyenkor rögzítik az autó állapotát és azt, hogy a kocsit milyen tartozékokkal adták át.

Például így:

A tulajdonos neve:	□□□□□□□□□□□□
a gk. forgalmi rendszáma:	□□□□□□□
sérült (igen/nem):	□□□□
rádió (van/nincs):	□□□□□
a rádió működik (igen/nem):	□□□□
pótkerék (van/nincs):	□□□□□
várható javítási költség:	□□□□□,□□

Ebben a képzeletbeli példában 7 különböző adat jellemez egy gépkocsit. Mivel a BASIC nyelvben nincs olyan változótípus, mely képes lenne az összetartozó 7 adatot egy egységként kezelni, a programozó kényszerűségből a 7 adatot külön-külön változókként írja le, és majd az algoritmus megfogalmazásakor vigyáz rá, hogy ezeket összetartozó egységként kezelje.

Például ilyen változókat deklarál:

```
2100 DIM A$(20): REM A TULAJDONOS NEVE
2100 DIM R$(6): REM A RENDSZÁM
.
.
2140 DIM M$(4) : REM A RÁDIÓ MŰKÖDIK
.
.
2160 K=0 : REM A JAVÍTÁSI KÖLTSÉG
.
.
```

Mivel a szervizben naponta nem csak egy gépkocsit vesznek fel, a változókat a programban úgy kell deklarálni, hogy a várható összes kocsi elférjen a gépben. Ha a szerviz egy nap — mondjuk — maximálisan 100 darab gépkocsit képes kiszolgálni, akkor ez lehet a megoldás:

```
.
2100 DIM A$ (20, 100) : REM A TULAJDONOSOK NEVEI
.
.
2160 DIM K(100) : REM A JAVÍTÁSI KÖLTSÉGEK
.
```

Itt most eltekintünk attól, hogy a BASIC-ben is van fájlkezelési lehetőség, ha olyan adattípus, mint a *fájl*, explicit módon nem is deklarálható.

Így az eredeti hét változóból 7, egyenként 100 elemből álló tömb lett. Az adatok azonban — sajnos — nem *gépkocsinként* vannak összefogva, ami a szerviz tevékenységét jobban tükrözné, hanem *nevenként, rendszámonként* stb. Ilyen egyszerű példánál nem is látszik még, hogy ez mekkora nehézségek forrása lehet később.

PASCAL-ban a dolog jóval logikusabb. A gépkocsira vonatkozó 7 darab *összetartozó* adatot egy rekordba foglalhatjuk. Így:

```
.
TYPE rádiótíp  =RECORD
                van: BOOLEAN;
                működik: BOOLEAN
                END;
gépkocsitíp    =RECORD
                tulajdonos: ARRAY [1..20] OF CHAR;
                rendszám: ARRAY [1..6] OF CHAR;
                sérült: BOOLEAN;
                rádió: rádiótíp;
                pótkerék: BOOLEAN;
                költség: REAL;
                END;
.
.
VAR gépkocsi: gépkocsitíp;
.
```

A programban aztán egy adott gépkocsi javítási költségére így hivatkozhatunk:

```
.
fizetendő:=gépkocsiköltség;
.
```

A PASCAL-ban van explicit módon deklarálható adattípus (a FILE — a fájl), mely a tömbbel (array) szemben a *végén nyitott*. Nem kell a programozónak azon töprengenie, hogy maximálisan hány gépkocsit képes fogadni egy nap a szerviz. Például így járhat el:

is, és amit elmondani szeretnénk ebben az olvasónaplószerű rovatban, több régebbi olvasmányélményhez is kötődik. A téma: a számítógép és a külvilág, az adatok és a valóság közötti kapcsolat. Minderről azért érdemes egy kicsit elmélkedni, mert a számítástechnika művelése megkívánja az absztrakciók tudatosítását: a gépben tárolt adatok ugyanis a külvilág valamilyen absztrakcióját képviselik.

```
VAR gépkocsi: gépkocsitíp;
    felvételek: FILE OF gépkocsitíp;
```

Lesz tehát egy „felvételek" nevű fájl, mely „gépkocsitíp" típusú rekordokból áll. A PASCAL nyelv beépített procedúráival a fájlba rekordok írhatók be, a fájlból a rekordok kiolvashatók, a fájl elejére lehet visszaállni stb.

A mondanivaló lényege: nagy jelentősége van annak, hogy az adatok struktúrája mennyire tükrözi az ábrázolni kívánt dolog struktúráját, és különösen fontos tudni, hogy a programozási nyelvek adatstrukturálási lehetőségei eltérőek (pl. a BASIC-é fejletlen, a PASCAL-é fejlett).

A fogalomalkotás és a mérési módszer

Ha most elszakadunk a gépkocsi-nyilvántartáshoz hasonló, triviális feladatok világától, és képzeletünk szárnyán a Galilei előtti korban lebegünk, érdekes lehet számunkra például az a jelenség, amelyet ma szabadesésnek nevezünk. A jelenséget (pl. egy ólomgolyó vagy egy tollpihe mozgását) szemlélve szembetalálkozhatunk a fogalomalkotás — a névadás — és a minősítés — mérési utasítás (mérési módszer) — kérdésével.

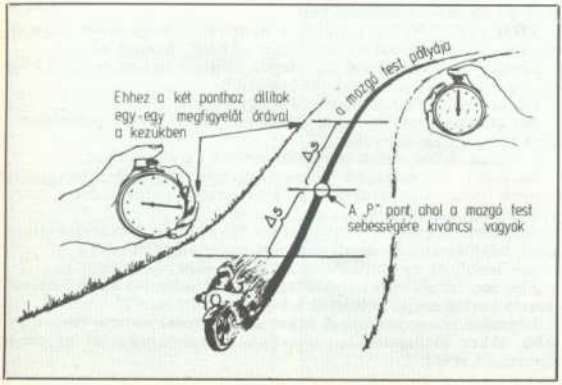
A fogalmak kiválasztása szorosan kapcsolódik a névadáshoz. Neveket vagy stabil, sokáig fennmaradó dolgoknak, vagy a dolgok olyan osztályainak adunk, amelyek kellő gyakorisággal jönnek létre ahhoz, hogy még akkor is rászolgáljanak névre, ha egyébként rövid életűek. A fogalmak helyes definíciója magába kell foglalja azt az utasítást is, ahogy a kérdéses fogalmat — legalább elvben — mérni lehet.

Ma már tudjuk, hogy a szabadesés leírásakor olyan fogalmakkal célszerű operálnunk, mint a megtett út (s), az eltelt idő (t), a pillanatnyi sebesség (v), a gyorsulás (a). A jelenség leírásakor ezen négy fogalom (mennyiség, mérőszám) között állapítunk meg törvényszerűséget.

A megtett út (s) mérési utasítása például ez lehet: „Vásárolj a boltban egy szabványos mérőszalagot, kezdőpontját helyezd az útszakasz elejére és olvasd le a mérőszalagról azt a számot, ami az útszakasz végével esik egybe".

A mozgó test pillanatnyi sebességének (v) már bonyolultabb a mérési utasítása és több lehetséges hibaforrást tartalmaz (lásd az ábrát). Például: „Az adott ponttól a pálya mentén mindkét oldalon jelölj ki egy-egy pontot kicsi (Δs) távolságra. Állíts mindkét ponthoz egy-egy megfigyelőt, kezükben egy-egy azonos időpontban elindított stopperórával. A két megfigyelő feladata az legyen, hogy amikor a tárgy előttük elhalad, megállítsák a kezükben levő stopperórát. Legyen a két stopperórán a megállításkor mért idő t_1 és t_2. A pillanatnyi sebesség adott „pontban" mért közelítő értékét így kapod meg:

$$v \approx \frac{2*\Delta s}{(t_2 - t_1)}$$



Minél kisebb Δs, az így számított érték annál jobban közelít a sebesség keresett pillanatnyi értékéhez. Csak elvileg, mert minél kisebb Δs, annál nehezebb pontosan meghatározni t_1 és t_2 különbségét (szemünk, reflexeink, a stopperórák pontatlansága annál jobban befolyásolja a mérést).

Ebben az példában az idő, az út, a sebesség és a gyorsulás fogalmát, nevét, mérési utasítását készen kaptuk, és ismerjük a rájuk vonatkozó összefüggéseket is, például zérus kezdősebesség esetén

$$s = \frac{a*(t_2 - t_1)^2}{2}\ ;\ v = a(t_2 - t_1).$$

De mit csináljunk akkor, ha számítógépünkkel új jelenséget vizsgálunk, új összefüggéseket kutatunk?

A világ új vagy eddig még nem vizsgált jelenségeinek strukturálása, új fogalmak alkotása és a névadás, megfelelő mérési utasítások kidolgozása a szellemi alkotó munka azon része, melyre tulajdonképpen nincs recept. A munkának ezt a részét a számítógépes feldolgozásoknál általában adottnak vesszük. Szinte kizárólagosan a már elvégzett mérések eredményeiből indulunk ki, melyek gyakran számok. Ez lényeges, mert a számítógép „szeret" számokkal dolgozni. Valahányszor mérésekkel adatokat generálunk a számítógép részére, tudatában kell lennünk, hogy azok *milyen típusú skálán* helyezhetők el. Nézzünk ennek a fontos kérdésnek a megvilágítására két mérést közelebbről.

(1) A képességvizsgálat szokásos gyakorlata, hogy a vizsgált személynek feladatokat adunk, és rendszerint minden egyes feladat megoldásáért valamilyen pontszám jár. „A képesség" fogalmához tehát tartozik egy mérési utasítás és a vizsgált személyek mindegyikéhez egy mérőszámot fogunk rendelni.

(2) A hálózati feszültség fogalmához is tartozik egy mérési eljárás („Dugd a feszültségmérő két pólusát a konnektorba és olvasd le a skálán a feszültség értékét"). Itt is egy számot kapunk, és a mértékegységet hozzátéve az eredmény mondjuk 218,3 V. E második esetben van értelme a következő típusú állításnak: a mérés időpontjában a hálózati feszültség 0,77%-kal volt alacsonyabb a névleges (220 V) értéknél.

Egészen más lehet a helyzet a képességvizsgálatnál. Ha az egyik vizsgált személy (A) történetesen 218,3 pontot ért volna el, míg a másik (B) 220 pontot, akkor jogos fenntartással kellene fogadnunk azt az állítást, hogy (A) képessége 0,77%-kal kisebb (B) képességénél. Még akkor is, ha a számítás kézenfekvő, ha a számítógép el is tudja könnyedén végezni az alábbi műveletet (mi az neki!?):

$$\frac{220 - 218{,}3}{220} \cdot 100 = 0{,}7727272$$

Mérési skálák

A számok (bárhonnan eredjenek is, bármilyen mérési utasítás alapján generálták is őket) arra csábítanak, hogy — különösen számítógéppel — műveleteket végezzünk velük: átlagot, szórást, arányokat stb. számítsunk belőlük. Tegyük ezt óvatosan: gondoljuk meg, hogy a továbbfeldolgozás alapjául szolgáló számok *„mennyit érnek"*, milyen skálán helyezhetjük el őket.

A „mért" és feldolgozandó (feldolgozásra csábító?) adatok az alábbi négy alaptípusba sorolhatók:

— Olyan adatok, melyek úgy keletkeznek, hogy megfigyeléseinket sorrendbe *nem* rendezett (nem rendezhető), de meghatározott kategóriába soroljuk. Például egy mozi bejáratánál figyeljük a

belépőket, és az adatokat úgy állítjuk elő (úgy „generáljuk" őket), hogy minden személyről megállapítjuk, visel-e kalapot. Külön számláljuk a „kalaposokat" és a „kalap nélkülieket". Azt a skálát, amelyben az ilyen típusú adatokat (a kalaposok és a kalap nélküliek számát) ábrázoljuk — amelyhez viszonyítjuk adatainkat — *nominális skálának* (angolul: nominal scale-nek) hívjuk.

— Olyan adatok, amelyek úgy jönnek létre, hogy megfigyeléseink eredményét valamilyen *sorba rendezett* kategóriákba soroljuk. Minden egyes kategóriáról meg tudjuk mondani, hogy nagyobb-e vagy kisebb-e a másiknál, de a kategóriák közötti *különbségre* nem vezetünk be mértéket. Így járunk el például, amikor egy csoport tagjait az iskolai végzettség szerint rendezzük. Az adatokat úgy „generáljuk", hogy mindenkitől megkérdezzük a legmagasabb iskolai végzettségét, az így nyert információt például az alábbi kategóriák szerint minősítjük: külön összeszámláljuk azokat, akik
=nem végezték el az általános iskolát sem;
=akik elvégezték az általános iskolát;
=akik valamilyen középiskolát végeztek;
=akik egyetemet vagy főiskolát végeztek.
Azt a skálát, amely szerint az ilyen adatokat elrendezzük, ún. *ordinális skálának* (angolul: ordinal scale-nek) nevezzük.

— Olyan esetekben, amikor az adatok nemcsak hogy sorba rendezhető kategóriákba sorolhatók, hanem már számszerűsíteni tudjuk a kategóriák közötti különbséget („távolságokat") is, *intervallumskálának* (angolul: interval scale-nek) nevezzük. Ilyen skálának nincs abszolút kezdő (zérus-) pontja. A kategóriákat pontszámtartománynak megfelelően jelöljük ki. Esetleg nem tudunk és nem is akarunk válaszolni arra, hogy egy-egy megszerzett pont „mennyit ér", nem is állítjuk, hogy a pontok értéke a skála mentén azonos lenne, és nincs értelme egy adott összpontszám alatt foglalkozni az eredményekkel; mert például nem igaz, hogy aki semmilyen kérdésre nem tudott értékelhető választ adni, az „abszolút tehetségtelen". Nincs értelme a kategóriákat az átlagos pontszámok *arányával* minősíteni: nem biztos, hogy akinek a pontszáma 200, annak kétszer nagyobbak a képességei, mint annak, akinek a pontszáma csak 100.

— A legkívánatosabb az a skála, amelynek van megfogható kiinduló (zérus-) pontja és amelyen kijelölhető egy világos, a skála egészére érvényes mértékegység. Az ilyen skálát *arányskálának* (angolul: ratio scale-nek) nevezzük. Ilyen skálán rendezhetjük például a hálózati feszültség időnkénti mintavételével nyert adatokat.

Valahányszor megfigyeléseink (méréseink) eredményét számszerűsítve adatokat „generálunk", tudatában kell lennünk annak, hogy milyen típusú skálát használunk. Arra nézve nincs pontos szabály, hogy mikor melyik a legmegfelelőbb skála. A skálákkal kapcsolatos döntésnek azonban jelentős kihatásai vannak a mérési adatok kiértékelési módszerére.

Ha egy megoldandó probléma figyelembevételével jól oldjuk meg az absztrakciót, ami által a valóságot a számítógép adataira transzformáljuk, nem hibáztuk el a „méréseket", a mérési eredményeket a megfelelő skálán rendeztük el, akkor általában keresni kezdjük a mérési adatok közötti *összefüggéseket*, azokat az oksági törvényszerűségeket, melyekre egy adott probléma megoldásánál szükségünk van: például kereshetjük az összefüggést a táplálkozási szokások és egyes megbetegedések, valamilyen szabálymódosítás és a balesetek számának alakulása stb. között.

A külvilágból nyert, gépbe vitt adatok és a számítógép elemeinek állapota közötti kapcsolat igen bonyolult. Ezeket a bonyolult kapcsolatokat az ember csak úgy tarthatja kézben, ha azokat hierarchikusan tagolja: a számítógép felhasználójához legközelebbi hierarchikus szintek egyike az a (rendszerint magas szintű) programozási nyelv, amelyen a programozó a megoldandó probléma ismeretében adattípusokat, konstansokat, változókat deklarál. Amikor ezt teszi, nagyon gyakran nincs tudatában annak az igen komoly absztrakciós műveletnek, amit végez. Reméljük, ez a cikk segít az absztrakciók tudatosításában.

(Lohnes és Codey: Introduction to Statistical Procedures with Computer Exercises, John Wiley and Sons, Inc. 1968.;
David Bohm: Okság és véletlenség a modern fizikában, Gondolat, 1960.;
Simonyi Károly: Villamosságtan, Akadémiai Könyvkiadó, 1954.;
Richard Dawkins: Az önző gén, Gondolat, 1986.)

— KE —

Integrál

Szövegszerkesztő I.

Ez a szövegszerkesztő program BASIC nyelven íródott, és 10-15 oldal szöveget tud egyszerre kezelni. Tartalmaz egy nyomtatóvezérlő rutint is, és így a legtöbb számítógépen jól szolgál. A többi szövegszerkesztőhöz hasonlóan az egyszerű írógépes szövegfeldolgozáshoz képest sok előnye van. Az egyik legjelentősebb, hogy lehetővé teszi *mondatok, szavak utólagos beszúrását* és törlését a szövegben. Megjegyzendő, hogy ez a blokk beszúrására már nem áll, mert egy blokk több mondatból épül fel, és külön blokkbetoldó utasítás nincs. További kényelem, hogy lehetséges a *blokkok mozgatása* a szövegben — például egy új bekezdés beszúrásához a szöveget a már begépelt anyag végére írjuk, és blokkmozgatással teszszük a helyére. Nagyon egyszerű a *folyamatos szövegbevitel* (hozzátoldás) is a már létező anyagokhoz.

Rátérve a szövegszerkesztővel folyó konkrét munkára, javasoljuk, hogy aki komolyan veszi a „lassan járj, tovább érsz!" bölcsességet, figyelmesen olvassa el az instrukciókat — miután a programot begépelte és rögzítette a kazettán vagy mágneslemezen —, majd üljön le a gép elé és kísérletezzen.

Az utasítások ismertetése után néhány egyszerű példát is közlünk, hogy segítsük a kezelés elsajátítását. Ezzel a kis gyakorlással a szövegszerkesztő kezelése nagyon egyszerűvé válik.

Nos, most vegyük úgy, hogy a program már kazettán van, és töltsük be a számítógépbe (CLOAD „SZERKESZ"). Indítsuk el: RUN. A képernyőn megjelenik a SZOEVEGSZERKESZTOE felirat *(1. ábra)*. Ezután a *főmenü* ugrik a képernyőre: a program mindig ide tér vissza az egyes funkciók befejezése után. A képernyő és a „papír", amelyre a nyomtatási kép készül majd, a könnyebb megkülönböztetés céljából eltérő formátumú. A menü képét a *2. ábrán* láthatjuk.

Túl azon, hogy kiválaszthatjuk a munkánk céljának megfelelő rutint, a képernyő további infomációkkal is szolgál: megadja az éppen használatos szövegfájl nevét és a szabad hely méretét.

Ha kiválasztottuk, hogy mit kívánunk tenni, és leütöttük a megfelelő betűt, részletesebb utasításokat is adhatunk. Vegyük például az F betűt (=formátum). Láthatóan most nem kell leütni az ENTER gombot ahhoz, hogy a gép elfogadja a választott funkciót. (Más esetekben, a villogó kurzorral beadott információk esetén mindig kell használni!)

Az F hatására a gép egy almenübe jut, ahol ellenőrizhetjük és módosíthatjuk a nyomtatáshoz szükséges paramétereket *(3. ábra)*. Ha az előző beállítás nem felel meg, egyszerűen csak át kell térni az új értékre. Ha az ENTER gombot lenyomjuk a változtatás bevitele nélkül, akkor a korábban közölt érték tárolódik a szöveggel és a programmal együtt. A következő adatokat lehet felfrissíteni: bal oldali margó, az első sor helye, sorhossz, sor/oldal, lapszám, lapdobás, az új bekezdés pozíciója, feloldás, fejléc. Ezek legtöbbje önmagáért beszél, de talán nem haszontalan, ha gyorsan végigtekintünk a választási lehetőségeken.

STOP (SZUENET). Leállítja a nyomtatót, hogy lapot tudjunk cserélni. A leállításnál a számítógép „BEEP" hangot ad.

Lapdobás. A következő lap elejére állítja a nyomtatót a szöveg leírása után (például új fejezet kezdődik).

Dupla szóköz. Ehhez nem kell magyarázat.

Bal oldali margó. A nyomtatófej bal oldali szélső pozíciójától mért távolság karakterekben.

Sorhossz. A bal oldali margótól mérjük karakterekben.

Bekezdés. A karakterek száma, amennyivel minden bekezdést beljebb írunk. A programnak a /p utasítással adjuk tudtára, hogy új bekezdés következik.

Oldalhossz sorokban. Ennyi sor fér egy oldalra: a nyomtató típusától, beállításától és az alkalmazott papír méretétől függ.

Sor/oldal. Az egy oldalra nyomtatni kívánt sorok száma.

Első sor. Az első sor helye a lap tetejétől számítva az első nyomtatásra kerülő sorig, beleértve a lapszámozást is.

Lapszám. Ha ez az érték Ø, akkor nem történik semmi. Ha 1 vagy több, akkor automatikusan növekszik minden oldalnál, és ezt a nyomtatás is tükrözi.

szoftver

```
        SZOEVEGSZERKESZTOE
 ================================

 I-IROOGEEP      F - FORMAATUM
  B-BETOLDAAS     K-KITOERLEES
 S-SZERKESZTEES
G - GLOBALIS SZERKESZTEES
 E-ELTAAROLAAS   V-VISSZATOELTEES
 M-MOZGATAAS     N-NYOMTATAAS
 U-UJRA
```

1. ábra

```
 ================================
 30,000 UERES HELY
AZ EERVENYES ADATAALLOMAANYNEEV
```

2. ábra

```
        MIT VAALASZT?
     OLDAL ELLENOERZEES
 ================================
 SZUENET (O=KI,1=BE) = 0
 LAPDOBAAS (O=KI,1=BE) = 1
 JUSZTIROZAAS (O=KI,1=BE) = 0
 KETTOES SZOKOEZ (O=KI,1=BE) = 0

BALOLDALI MARGOO HELYE 10
 SORHOSSZ 60
 /p BEKEZDEES HELYE 0
 OLDALHOSSZ SOROKBAN= 66
 SOR/OLDAL= 54
 KEZDOEDIK A  5 -IK SORBAN
 OLDALSZAAM= 0



FEJLEEC=
```

3. ábra

```
        MIT VAALASZT?
     SZOEVEGBEADAAS
 ================================
        PARANCSOK:
 /f   UJ OLDAL
 /aN   ASCII-KODOK
 /c   POZICIONAALAAS
 /i   MARGOO BEAALLITAAS
 /p   UJ BEKEZDEES
 /sN  UGORJ AZ N#-IK SORRA
 /tN  TABULAATOR POZICIOO
 /l   UJ SOR
 /u   UJRA
  e   A BEADOTT SZOEVEG SZERKESZ
TEESE
 ================================
   reegi, VAGY uj
 KEESZ (I/N)
```

4. ábra

5. ábra

```
   <I>-BETOLDAAS <C>-CSERE <D>-T
OERLEES
 i/lalma
  # 1



        SZOEVEGSZERKESZTOE
 ================================

 T - TIPUS       F - FORMAATUM
  B-BETOLDAAS     K-KITOERLEES
 S-SZERKESZTEES
G - GLOBALIS SZERKESZTEES
 E-ELTAAROLAAS   V-VISSZATOELTEES
 M-MOZGATAAS     N-NYOMTATAAS
 U-UJRA


 ================================
 29,967 UERES HELY
AZ EERVENYES ADATAALLOMAANYNEEV
```

```
        MIT VAALASZT?
BETOLDAAS
  # SORT, u, VAGY ''

        SOR KERESEES
 ================================
```

6. ábra

```
 3

 4

 3

 2   barack

 AZ  3 -IK SOR ELEE

  # SORT, u, VAGY ''

        SZOEVEGSZERKESZTOE
```

7. ábra

```
        MIT VAALASZT?
     TOERLEES

  # SORT, u, VAGY ''


  3
 TOERLI EZT A SORT
  # SORT, u, VAGY ''
```

8. ábra

```
 ================================
 29,979 UERES HELY
AZ EERVENYES ADATAALLOMAANYNEEV


        MIT VAALASZT?
     SZERKESZTEES
 ================================
# SORT, u, VAGY ''

        SOR KERESEES
 ================================

 2

 3

 2
```

9. ábra

```
<I>-BETOLDAAS <C>-CSERE <D>-TOE
RLEES
        SZERKESZTEES
 ================================









   <I>-BETOLDAAS <C>-CSERE <D>-T
OERLEES
        BETOLDAAS
```

10. ábra

11. ábra

```
   <I>-BETOLDAAS <C>-CSERE <D>-T
OERLEES
        TOERLEES

        TOERLEES

        TOERLEES

        TOERLEES

        TOERLEES
```

Fejléc. Csak akkor kérhetjük a megjelenítését, ha a lapszámot is nyomtatjuk (például „MIKROMAGAZIN, 1987. február").

Szöveg beírása

Ezt a funkciót (a főmenüben I-írógépfunkció; lásd a 2. ábrát) új szöveg bevitelére és egy már létező szövegfájl utólagos kiegészítésére használhatjuk. A program megkérdezi, hogy „új" vagy „régi" szöveggel dolgozunk-e. Az „új" megadása törli a régi szöveget a memóriából (a szalagról nem), a „régi" megadása a pointert a memóriában lévő szöveg végére állítja.

```
   <I>-BETOLDAAS <C>-CSERE <D>-T
OERLEES
        CSERE

        SZERKESZTEES
 ================================

toekoek/dinnye
```

12. ábra

```
        MIT VAALASZT?GLOBALIS S
ZERKESZTEES
 ================================
u, VAGY AALLOMAANYKERESEES>
BEADANDO AZ UJ>
```

13. ábra

```
        MIT VAALASZT?
 AALLOMAANYELTAAROLAAS
 ================================
 EERVEENYES AALLOMAANYNEEV=
 AALLOMAANYNEEV>
```

14. ábra

15. ábra

```
    A MOTOR BEKAPCSOLVA

    A MOTOR BEKAPCSOLVA

    A MOTOR BEKAPCSOLVA

    A MOTOR BEKAPCSOLVA

    A MOTOR BEKAPCSOLVA

      ELTAAROLAASALMA

        SZOEVEGSZERKESZTOE
```

```
        MIT VAALASZT?
 BLOKK MOZGATAAS
 ================================
 BEADANDOO
 <B>  A BLOKK ELEJE
 <E>  A BLOKK VEEGE
 <T>  JELZOE
 KEESZ

        SOR KERESEES
 ================================

 0

 1

 BLOKK MOZGATAAS
 ================================
A BLOKK ELEJE= 0
 A BLOKK VEEGE=  0
JELZOE=          0
 TILTOTT MOZGATAAS
```

16. ábra

17. ábra

```
        MIT VAALASZT?
 SZOEVEGNYOMTATAAS
 ================================
AZ OLDAL FORMAATUMA RENDBEN ?
 KEZDEES HELYE
FORMAAL JA?
 SOROS, VAGY PAARHUZAMOS NYOMTAT
OO      (s OR p)?

 NYOMTATAAS
 ================================
```

A szövegbeviteli funkció esetén a gép felszabadítja a billentyűket és alaphelyzetben kisbetűket írhatunk a képernyőre, illetve a memóriába. A SHIFT gomb lenyomásával kapjuk a nagybetűket, ahogy ez szokásos. Ebben az esetben az utasításokat kisbetűkkel kell beírni, és mindig az előtt a mondat előtt, ahol a hatást érvényesíteni kívánjuk.

- /f — Új oldalra (valószínűleg senki sem fogja használni, míg nem látta a szöveg első példányát papíron).
- /aN — ASCII kódokat küld a nyomtatóhoz 0—255-ig. Egymás után annyi adható meg, amennyi a munkához szükséges. Ezekkel a karakterekkel lehet a nyomtatón például apróbetűt vagy vastagon szedett szöveget előállítani. Az N aktuális értékei megtalálhatók a nyomtatók kézikönyvében.
- /c — Lapközépre teszi a szöveget, például a címeket, alcímeket.
- /i — Mindkét margót öt karakterrel beljebb viszi addig, míg másodszor meg nem adjuk az /i utasítást.
- /p — Soremelés után új bekezdés indul. Ha megadtuk, hogy hány hellyel kezdődjön beljebb a szöveg, a nyomtatásban ez érvényesülni fog. Ne tévesszük össze a /S1 utasítással, amely egy soremelést eredményez.
- /SN — N számú, max. 99 soremelést hajt végre. Ha mondjuk csak hat sort akarunk emelni, és a szövegben az első karakter egy szám, akkor a program az első számjegyet még a soremelés részeként értelmezheti. Ezért célszerű az első karakter előtt egy szóközt beiktatni, vagy a számot kétkarakteres formában megadni: például 06.
- /tN — Tab utasítás, N a lépések száma.
- /l — Új sor. Akkor kell, ha például címzést írunk.
- /u — Befejezte, újra a főmenübe térünk vissza.

Az utasításokat szabad láncolni egy sorba, de minden utasításnak a mondat előtt kell állnia. Lássunk egy példát:

a
/f/Sl0/c MIKROMAGAZIN
utasítássor hatása a következő lesz: új oldalon 10 soremeléssel, továbbá annyi soremeléssel, amennyit az első sorra megadtunk a nyomtatási funkcióknál, kerül a /c hatására középre a MIKROMAGAZIN szöveg.

Szövegszerkesztővel legalább olyan gyorsan lehet gépelni, mint írógépen, különösen akkor, ha gyakran ejtünk gépelési hibákat. A szövegbeli elütéseket a BACKSPACE billentyűvel (balra nyíl←) egyszerűen törölhetjük, mielőtt az ENTER gombot lenyomnánk. ENTER után a szöveg már a tárba kerül, és csak a szerkesztő üzemmódban lehet korrigálni a hibákat. ("e" és ENTER gomb lenyomására szerkeszthető az előző mondat.) Ilyenkor a hibákat ideiglenesen hagyjuk benn a legközelebbi javításig, például amíg a munka lezárásakor átnéztük a szöveget.

A szerkesztési üzemmód az "S" beírásával választható a főmenüből. Amikor elérjük a mondat végét, nyomjuk le az ENTER gombot. Ha nincs hibajelzés, írhatjuk tovább a szöveget.

Beszúrás, törlés, blokkmozgatás

A szöveget a begépelés után célszerű ismét átnézni. Az ekkor felfedezett hibákat a *sorkereső* segíti kijavítani (lásd a sorkeresést, 9. ábra). A képernyőn megjelenik a kérdés:
SOR#, U VAGY " "?

A következő kínálatból választhatunk: kiindulhatunk a keresni kívánt mondat sorszámából (SORSZÁM), visszatérhetünk a menübe (U), vagy kereshetünk egy konkrét szövegezésű mondatot (erre utalt a kérdésben az idézőjel). Az üresen, adatbeadás nélkül beütött ENTER jelenti a harmadik esetet, és ezután már dolgozhatunk a *sorkereső* rutinnal. A szövegen végig lehet mozogni a *fel, le* nyilak segítségével (ezek a billentyűk funkcióismétlők, ha lenyomva tartjuk őket). A betűváltó gombbal (SHIFT) együtt lenyomva egyszerre nyolc mondatot ugrunk. A mondat sorszáma a szöveg bal oldalán található. Az ENTER gomb lenyomásával tudatjuk a programmal, hogy a keresett szöveget megtaláltuk. Ha egy új mondatot beszúrunk vagy törlünk, a mondatok számozása automatikusan megváltozik. Javítás vagy törlés után ismét a *sorszám újra* vagy a *sorkereső* opcióval találkozunk. Az ENTER lenyomásával újra mozoghatunk a nyilakkal a szövegben.

Betoldás a főmenü B funkciója (lásd az *5—7. ábrát*). A lehetőség egy mondat bevitelére vonatkozik, amely a kiválasztott mondat elé lesz betoldva. A program visszajelzi a helyet, és vár az igenlő válaszra (i), hogy a betoldást elkezdjük. Ha nemleges a válasz, újra kérdez a helyre stb.

Törlés a főmenü K funkciója (lásd a *8. ábrát*). A program megkérdezi: „töröljem ezt a sort?", igenre töröl, nemnél nincs változás. Ha sok a törölni kívánt szöveg, a szöveg vége felől haladjunk előre; a gép nem veszteget időt az újraszámozásra.

A *szöveg mozgatása* a főmenü M funkciója (lásd a *16. ábrát*). A sorkereső segítségével gyorsan kijelölhetjük a mozgatandó szövegrészlet elejét és végét, valamint azt a helyet, ahová tenni akarjuk. A képernyőn a következő utasítások jelennek meg:
B szövegrészlet *kezdete*
E szövegrészlet *vége*
T cél, itt fog *kezdődni*
Sorkereső <ENTER> vagy <U><ENTER> az újraindításhoz.

B benyomásával megjelöljük a mozgatni kívánt szövegrészlet elejét, *E* lenyomásával az utolsó mozgatni kívánt mondatot. *T* megjelöli azt a helyet, amely elé akarjuk tenni a szövegrészletet. Miután mindhárom pontot megjelöltük, nyomjuk le az ENTER gombot, és ha mindent jól csináltunk, másodperceken belül visszatérünk a menühöz. Hosszú blokkok mozgatása kicsit tovább tarthat. Ha a mozgatás koordinátáinak nincs értelme, például a kezdetre megadott helyzet magasabb sorszámú mondatnál van, mint a szövegrészlet vége, <TILOS MOZGATÁS> hibajelzést kapunk.

A szöveg javítása után újra nézzük át az elkészült anyagot! Ez megtehető a főmenü "S" (szerkeszt) és a "G" (globális szerkesztés) funkcióiból. Az utóbbinál lehetőség van rá, hogy egy szót vagy kifejezést keressünk ki és cseréljünk.

A *sorkeresővel (9. ábra)* megkereshetjük a kívánt szövegrészletet. Ekkor nyomjuk le az ENTER gombot; hatására a kiválasztott mondat a munkaterületre kerül. Ha tudjuk, hogy melyik mondatot kívánjuk cserélni, a *sorkereső* kérdésére közvetlenül a mondat száma is beadható.

A kiválasztott mondat egy speciális szerkesztési képernyőre kerül *(10. ábra)*. A képernyőn felül a SZERKESZTEES felirat látható. A kurzor a szokásos nyilakkal mozgatható a mondaton belül. Ebben az üzemmódban csak akkor lehet kilépni a mondatból, ha BESZÚRÁS vagy CSERE módot választunk. A billentyűzetnek kisbetű üzemben kell lennie.

TÖRLÉS, CSERE, BETOLDÁS üzemmód esetén állítsuk a kurzort az elé a karakter elé, ahol a változtatást akarjuk, és nyomjuk le a <T>OERLEES <C>SERE <B>ETOLDAAS gombokat.

TÖRLÉS. Ahányszor lenyomjuk a <D> gombot, a kurzor jobb oldalán annyi karakter „tűnik el" *(11. ábra)*. A <D> gomb folyamatos lenyomásával hosszabb karaktersorozatot törölhetünk. Bármely más billentyű lenyomása a SZERKESZTÉS mint képernyőüzemmód visszatérését eredményezi.

CSERE. A <C> lenyomására a kurzor villogni kezd *(12. ábra)*. Ekkor a szöveg felülírható. ENTER-rel lehet visszatérni a SZERKESZTÉS alaphelyzetbe. A változtatás eredménye belekerült a mondatba.

BETOLDÁS. <B> lenyomásakor a kiválasztott mondat a kurzornál „kettészakad", hogy legyen hely a szöveg beírására. Szintén az ENTER-rel jutunk vissza a SZERKESZTÉS alaphelyzetbe.

CSERE és BETOLDÁS esetén a balra nyíl (visszaléptetés) normálisan működik, de a kurzor egyéb mozgatásához a SZERKESZTÉS üzemmódban kell lenni. Az ilyenkor beadott „üres" ENTER a *sorkereső* megjelenését eredményezi.

GLOBÁLIS SZERKESZTÉS. <G> segítségével a kívánt részletet megkereshetjük a szövegben *(13. ábra)*, megnézhetjük az eredeti szöveget, és hogy milyen lesz a javítás után. A „szerkesztő" megkérdezi, hogy minden rendben van-e, vagy további javítás szükséges. "n" beírására várni fogja a változtatást. "U" megadására visszatér a menühöz. Ha csak az ENTER gombot nyomjuk le, akkor az új szöveget elhelyezi a régi helyén.

KERESÉS. Ha az ENTER gomb lenyomása után a szövegmezőben csak "f"-et talál a program ("f" és semmi más), akkor kilép az eddigi üzemmódból, de a SZERKESZTÉS változatlanul rendelkezésre áll. Ezt a rutint akkor használhatjuk, ha valamely szövegrészletet ki akarjuk cserélni, de nem tudjuk pontosan, hogy hol van.

TÁROLÁS. A főmenü E funkciója (lásd a *14. ábrát*). Az elkészült szöveg kazettára (mágneslemezre) vitelére szolgál. Figyeljünk rá, hogy a kazettán nehogy véletlenül felülírjunk egy már létező szöveget. Akárcsak a programok tárolásánál, ügyeljünk a helyes pozicionálásra.

Célszerű a szöveget valamilyen fájlnévvel ellátni, ami maximum 8 karakter hosszú lehet. A magnót állítsuk felvétel helyzetbe — erre a program villogva figyelmeztet, lásd a *15. ábrát* —, majd nyomjuk le az ENTER gombot. Amikor a rögzítés kész, a főmenü tér vissza a képernyőre.

VISSZATÖLTÉS. A főmenü V funkciója. Kiválasztásához beütjük a V betűt és a fájl nevét. A kazettán álljunk a megfelelő helyre a számláló segítségével. Ha nem tudjuk hol van, célszerű a kazetta elejére állni. Nyomjuk le az ENTER gombot. A gépben lévő régi szöveg törlődik, mielőtt az új betöltődik. A visszatöltés befejezésekor ismét a főmenü jelenik meg a képernyőn. Ha nem akarjuk a tárolást vagy a visszatöltést végrehajtani, akkor a fájlnév helyett "u" beírására egyszerűen visszatérünk a főmenübe anélkül, hogy a szöveg megsérült volna.

Nyomtatás

Ez a főmenü N funkciója (lásd a *17. ábrát*). Ha a nyomtató kódokkal vezérelhető (C.ITOH, EPSON, COMPUMATE, TERTA, ROMON stb.), akkor az írássűrűséget, betűtípust meghatározó kódokat célszerűen a szöveg nyomtatása előtt kell kiválasztani ASCII kódok formájában. A nyomtatót vezérlő menü megkérdezi a nyomtatás beállításához szükséges információkat (lásd a 4. ábrát). Ha az 1. ábrának megfelelően a STOP-ot bekapcsoltuk, és A4-es formátumú írógéppapírra elindult a nyomtatás, a nyomtató minden lap végén leáll, és vár a papírcserére. A program kiírja a nyomtatás alatt lévő oldal és mondat sorszámát. A nyomtatás "U"-val bármikor leállítható.

Kilépés a szövegszerkesztőből

A főmenü "U" funkciójával kilépünk a szövegszerkesztőből, és újra visszatérünk a számítógép BASIC interpreterjéhez.

Végezetül javasoljuk, hogy miután egy-két oldal szöveget bevittünk a gépbe, tároljuk el kazettára és próbáljuk ki a funkciók működését. Különösen a SZERKESZTÉS funkciót célszerű begyakorolni. Változtassuk meg a bekezdések helyét (/i), vagy hozzunk középre egy kifejezést (/c) stb. A gyakorlás megkönnyítésére néhány további információt és példát közlünk a következő számunkban.

GALINA FERENC

THE USERS PORT

San Fernando Valley Commodore Users Group

C64 programok átírása C128-ra

Az alábbi négy táblázattal átírhatók az olyan C64-re írt programok, amelyek PEEK, POKE, SYS utasításai csak az első három lap címeit használják. Ez egyébként elég gyakori eset. A táblázatok az eredeti és az új címeket decimális és hexadecimális formában egyaránt tartalmazzák.

SIMONYI ZSUZSA

Az eredeti cikket jól kiegészíti egy másik klub, a Pittsburgh Commodore Group *PCG Newsletter* c. lapjának ugyanezzel a témával foglalkozó írása. Eszerint a C128 az ún. 128-as üzemmódban néhány, a C64-nél is létező utasítást másként értelmez:

- FREE — az argumentum a memória lapsorszámát jelzi
- LIST — nem működik program módban
- RESTORE — sorszámmal egészül ki
- SYS — a regisztereknek lehet értéket adni

Ezenkívül figyelembe kell venni, hogy számos új kulcsszó is létezik. (A szerk.)

C-64 DEC/HEX	C-128 DEC/HEX	C-64 DEC/HEX	C-128 DEC/HEX
0003/0003	4476/117C	0047/002F	0049/0031
0004/0004	4477/117D	0048/0030	0050/0032
0005/0005	4474/117A	0049/0031	0051/0033
0006/0006	4475/117B	0050/0032	0052/0034
0007/0007	0009/0009	0051/0033	0053/0035
0008/0008	0010/000A	0052/0034	0054/0036
0011/000B	0013/000D	0057/0039	0059/003B
0012/000C	0014/000E	0058/003A	0060/003C
0013/000D	0015/000F	0059/003B	4608/1200
0014/000E	0016/0010	0060/003C	4609/1201
0016/0010	0018/0012	0061/003D	0062/003E
0017/0011	0019/0013	0065/0041	0067/0043
0019/0013	0021/0015	0066/0042	0068/0044
0020/0014	0022/0016	0067/0043	0069/0045
0021/0015	0023/0017	0068/0044	0070/0046
0034/0022	0036/0024	0069/0045	0071/0047
0035/0023	0037/0025	0070/0046	0072/0048
0036/0024	0038/0026	0071/0047	0073/0049
0037/0025	0039/0027	0072/0048	0074/004A
0043/002B	0045/002D	0073/0049	0075/004B
0044/002C	0046/002E	0074/004A	0076/004C
0045/002D	4624/1210	0077/004D	0079/004F
0046/002E	4625/1211	0078/004E	0080/0050

1. táblázat

C-64 DEC/HEX	C-128 DEC/HEX	C-64 DEC/HEX	C-128 DEC/HEX
0174/00AE	0174/00AE	0214/00D6	0235/00EB
0175/00AF	0175/00AF	0511/01FF	0511/01FF
0178/00B2	0178/00B2	0512/0200	0512/0200
0179/00B3	0179/00B3	0601/0259	0866/0362
0183/00B7	0183/00B7	0611/0263	0876/036C
0184/00B8	0184/00B8	0621/026D	0886/0376
0185/00B9	0185/00B9	0631/0277	0842/034A
0186/00BA	0186/00BA	0641/0281	2565/0A05
0187/00BB	0187/00BB	0642/0282	2566/0A06
0188/00BC	0188/00BC	0643/0283	2567/0A07
0193/00C1	0193/00C1	0644/0284	2568/0A08
0194/00C2	0194/00C2	0646/0286	0241/00F1
0195/00C3	0195/00C3	0649/0289	2592/0A20
0196/00C4	0196/00C4	0650/028A	2594/0A22
0197/00C5	0212/00D4	0653/028D	0211/00D3
0198/00C6	0208/00D0	0663/0297	2580/0A14
0200/00C8	0234/00EA	0780/030C	0006/0006
0201/00C9	0232/00E8	0781/030D	0007/0007
0202/00CA	0233/00E9	0782/030E	0008/0008
0203/00CB	0213/00D5	0783/030F	0009/0009
0208/00D0	0214/00D6	0784/0310	4632/1218
0211/00D3	0236/00EC	0785/0311	4633/1219
0213/00D5	0231/00E7	0786/0312	4634/121A

3. táblázat

C-64 DEC/HEX	C-128 DEC/HEX	C-64 DEC/HEX	C-128 DEC/HEX
0079/004F	0081/0051	0115/0073	0896/0380
0084/0054	0086/0056	0122/007A	0061/003D
0084/0054	0087/0057	0123/007B	0062/003E
0085/0055	0088/0058	0139/008B	4635/121B
0090/005A	0092/005C	0140/008C	4636/121C
0091/005B	0093/005D	0141/008D	4637/121D
0095/005F	0097/0061	0142/008E	4638/121E
0096/0060	0098/0062	0143/008F	4639/121F
0097/0061	0099/0063	0144/0090	0144/0090
0098/0062	0100/0064	0145/0091	0145/0091
0099/0063	0101/0063	0147/0093	0147/0093
0100/0064	0102/0066	0148/0094	0148/0094
0101/0065	0103/0067	0149/0095	0149/0095
0102/0066	0104/0068	0151/0097	0151/0097
0104/0068	0105/0069	0152/0098	0152/0098
0105/0069	0106/006A	0153/0099	0153/0099
0106/006A	0107/006B	0154/009A	0154/009A
0107/006B	0108/006C	0157/009D	0157/009D
0108/006C	0109/006D	0160/00A0	0160/00A0
0109/006D	0110/006E	0161/00A1	0161/00A1
0110/006E	0111/006F	0162/00A2	0162/00A2
0111/006F	0112/0070	0163/00A3	0163/00A3
0112/0070	0113/0071	0165/00A5	0165/00A5

2. táblázat

C-64 DEC/HEX	C-128 DEC/HEX	C-64 DEC/HEX	C-128 DEC/HEX
0788/0314	0788/0314	0805/0325	0805/0325
0789/0315	0789/0315	0806/0326	0806/0326
0790/0316	0790/0316	0807/0327	0807/0327
0791/0317	0791/0317	0808/0328	0808/0328
0792/0318	0792/0318	0809/0329	0809/0329
0793/0319	0793/0319	0810/032A	0810/032A
0794/031A	0794/031A	0811/032B	0811/032B
0795/031B	0795/031B	0812/032C	0812/032C
0796/031C	0796/031C	0813/032D	0813/032D
0797/031D	0797/031D	0814/032E	0814/032E
0798/031E	0798/031E	0815/032F	0815/032F
0799/031F	0799/031F	0816/0330	0816/0330
0800/0320	0800/0320	0817/0331	0817/0331
0801/0321	0801/0321	0818/0332	0818/0332
0802/0322	0802/0322	0819/0333	0819/0333
0803/0323	0803/0323	0828/033C	2816/0B00
0804/0324	0804/0324	/ /	

4. táblázat

KI AD MAGYARÁZATOT?

ATARI-BASIC. Egy programmal SAVE paranccsal vigyük háttértárolóba, majd LOAD (vagy CLOAD) parancs segítségével töltsük vissza. Ezt ismételjük meg. Azt tapasztaljuk, hogy parancsunk mindig hosszabb lesz. Miért?

Tóth József

Az idén negyedszer került sor Szekszárdon az *országos* pályázat döntőjére. Az NJSZT, a *Garay* Gimnázium és a lap szerkesztősége által meghirdetett *pályázatra* a viszonylag rövid határidő ellenére is 76 pályamű érkezett, melyek közül csak egynek volt két szerzője, viszont többen 2—4 programmal jelentkeztek. Összesen 55 oktató- és 42 játékprogram futott be. A kiírás oktató- és játékprogram kategóriában a legfeljebb elsőéves egyetemistáknak szólt. A pályaműveket Koltai Márta és Siegler Gábor tanárok vezetésével a budapesti Berzsenyi Dániel Gimnáziumban lapunk újjáéledő diákszerkesztőségének tagjai előzetesen elbírálták, majd a Garay Gimnázium, az NJSZT, a KISZ KB KSZT és lapunk képviselőinek jelenlétében valamennyit bemutatták. Végül az oktató kategóriában 11, a játékban pedig 8 program került a döntőbe, melyet Szekszárdon, a Garay Gimnáziumban március 19-én és 20-án tartottunk meg. *(1. kép)*

A bírálati szempontok

— A versenyző a pályaművet szórakoztatóan, gördülékenyen, szellemes előadással mutassa be a közönségnek. *(2. kép)*

— A programot ne lehessen könnyen elrontani („bolondbiztos" legyen).

— Segítse elő, könnyítse meg a vele való munkát.

— A magyar helyesírás szabályait, a magyar betűkészletet alkalmazza.

— Közérdekű (közérdeklődésre számottartó) problémával foglalkozzon az oktatóprogram.

— A játékban az adott géptípus maximális képességeit érvényesítse a program alkotója.

A döntős programok a közönség élénk érdeklődésétől kísérve *(3. kép)* két napon keresztül futottak a játék- és oktatóteremben felállított gépeken, miközben a 11 tagú zsűri egy másik helyiségben egyenként értékelte a szerzők általi programbemutatást. 19-én az oktató-, 20-án a játékprogramok kerültek sorra.

Az élmezőny

Az *oktatóprogram kategóriában* első díjat nyert *Nagy Ákosnak*, a BME elsőéves hallgatójának *(4. kép)* Pascal nyelven ZX-Spectrumra írt biológiai: génekkel, génsebészettel, környezeti hatásokkal foglalkozó szimulációs programja. Képzeljük el, hogy a tanórán belül egy-egy konkrét egyszerű szervezetet (egysejtűt) alapvető génjeivel jellemezve kísérleti alanyként tekinthetünk. A program hatféle környezeti feltétel változtatásával és az utódok véletlenszerű vagy művi keresztezésével halad generációkon át; minden generációnál vizsgálhatjuk a tulajdonságok *változásait*, sőt a környezeti feltételek megváltoztatásával az előzőleg még életképes egyedek utódairól kiderülhet, hogy azok már csak vegetálnak, míg az előzőleg kedvezőtlen perspektivájú tenyészetek „följöhetnek" az új körülmények között.

A lapunk által felajánlott TVC (VIDEOTON) számítógépet a zsűri elnöke, Hajós Lajosné vezető rendszerszervező adta át.

1. kép

2. kép

Második díjat, 2000 forintos vásárlási utalványt érdemelt *Kövesi Zsolt*, a bátaszéki gimnázium tanulója *(5. kép)* HT gépre írt, a nagyított orosz ábécét ismerő, az orosz nyelv oktatását segítő programjával.

Harmadik díjas lett *Pulay Gábor*, a balassagyarmati Balassi Bálint Gimnázium tanulója *(6. kép)* C Plus/4-re írt, kétváltozós függvényeket ábrázoló programjával. 1000 forintos vásárlási utalványt nyert.

A *játékkategória* első díjasa a tavalyi második helyezett: *Rátkai István*, a budapesti Fazekas Mihály Gimnázium diákja *(7. kép)*. C64-re készített, „Időrégész" kalandjáték-programjával a KISZ KB Középiskolai és Szakmunkástanuló Tanácsa díját, egy Primo A32-es számítógépet nyert.

Ez már igazi kaland! Több mint 50 romantikus, múltidéző helyszínen bolyonghatunk —

5. kép

3. kép

4. kép

I vitázva

6. kép

hogy fölleljük a mesebeli griffmadár tojását. Őserdőben, faoduban, feneketlen kútban, kísértetkastélyban, sziklaszakadékban stb., s mindeközben száz és ezer veszéllyel, illetve turpissággal találkozva élvezhetjük a színes vándorlást, élesíthetjük számos szellemi képességünket. Ehhez a program kétszáznyi, ragozott alakot is megértő szótára támogat az eligazodásban, sőt, segítségkérésre tanácsot is ad (például: „Csereberélj a boszorkánnyal!"). Mivel a játék nagyon bonyolult, egy-egy izgalmas helyzetből — az állapotot kimentve — sok variáns is lejátszható, ami több napig is tarthat.

Második lett a tavalyi különdíjra érdemesült *Pallagi László*, az érdi Vörösmarty Gimnázium volt növendéke *(8. kép)* HT gépre írt logikai játékprogramjáért. Egy játékossal szemben a program a partisorozatban gyorsan fejlődik (tanul). Kezdetben „mulya", de rövidesen méltó ellenfélé válik. A díj 2000 forintos vásárlási utalvány volt, ami mellé biztosan jól jött a jelenlegi kiskatonának az NJSZT 1800 forintos különdíja, melyet szintén elnyert szépen strukturált, gyors, sok funkciójú — pusztán csak BASIC! — programjával.

Harmadikként, 1000 forintos vásárlási utalvánnyal jutalmazta a zsűri *Balázs Róbertet*, aki szintén a balassagyarmati Balassi Bálint Gimnáziumba jár *(9. kép)*. A C64-re alkotott „Ninja" karateprogramhoz még az NJSZT 1500 forintos

7. kép

prémiumát is megkapta a jól kidolgozott funkciójavító segédprogramokért.

A második és harmadik díjakat a nyertesek a Novotrade-nek köszönhetik.

• És akik még kiválóak

A különdíjazásra a Tolna Megyei Tanács (5000 Ft), a megyei NJSZT (3000 Ft), a Garay Gimnázium KISZ Bizottsága (2000 Ft) és az NJSZT Ifjúsági Bizottsága (1800 Ft) részéről befolyt összeget fordíthatta a zsűri. Ebből 2000 forintot kapott a mezőny legfiatalabb tagja, *Békési Gábor*, aki általános iskolás létére nagyszerű fantáziával választotta meg témáit, és mindkét kategóriában bekerült a legjobbak közé *(10. kép)*. Játékában a képtár éke, a „Ketyegő madonna" elrablójának mozaikképét kell „kijátszania" a tanúnak (a játékosnak); a „Konverkli" kezelőjének, tekerőjének (a tanulónak) pedig akár Einsteinnel is vitázva kell döntenie a különböző számkonverzációs megoldások között.

8. kép

9. kép

Győrfi Gábort, a budapesti Képző- és Iparművészeti Szakközépiskola tanulóját *(11. kép)*, aki nagy vitát kavaró, a zsűritagok által is szélsőségesen értékelt (3—10 pontos) „Ember és kenyere" című programjával új műfajt teremtett, 1500 forintos díjjal tüntették ki. A megtanulandók közlése után ezek besúlykolását a tanulási egységek közé célszerűen belekomponált játékprogramokkal segítette elő, pontosabban: addig nem haladhat a tanuló tovább, amíg a „vizsgajátékban" nem győz.

A szintén Bátaszékről való *Bomba Gábor* pályaműve C64-en a német alapszókincs elsajátítását, valamint a legfontosabb nyelvtani összefüggések gyakorlását támogatja — 1000 szavas szótárral, játékosan, kisgyerekek számára vonzóan segíti. 1000 forintos különdíjban részesült.

Végül: 1000—1000 forintos vásárlási utalvánnyal ismerték el a bírálók *Négeli Péternek*, a szek-

10. kép

11. kép

szárdi Garay Gimnázium tanulójának teljesítményét, és természetesen a játékkategória nyertesét, *Rátkai Istvánét*.

Az eredményhirdetést követően a verseny házigazdája, Zentay András igazgató bejelentette, hogy jövőre ismét meghirdetik a pályázatot, amelynek színvonala a zsűri egyöntetű véleménye szerint örvendetesen magasabb volt a tavalyinál.

A látványos és ötletgazdag alkotások nagy közönségsikert arattak március 24-én, a II. Országos Mikroszámítógépes Találkozó keretében, a BNV 25-ös pavilonjában megtartott bemutatón is. 11 órától 16 óráig hat gépen egymás mellett futottak a programok. Konzulensként a két országos bajnok állta a sarat. Gondoljuk el, hogy ez nem is volt könnyű: több száz fiatalt fogadtak, és sok felnőtt is kíváncsi volt a részletekre.

BITEK ÉS FIGURÁK

Bástyavégjátékok

A középjáték bonyodalmai közben — éppen kisebb értékük miatt — először általában a könnyű tisztek hullanak el, de igen gyakori a vezérek korai cseréje is. Legkésőbb a bástyák kapnak szerepet a játékban. Ahhoz ugyanis, hogy aktívan beavatkozzanak a küzdelembe, nyílt vagy legalább félig nyílt vonalra van szükségük. Ez azonban hosszú előkészítést igényel — miközben a többi tiszt lecserélődik —, és így áll elő a végjáték, amelyben bástya vagy bástyák és gyalogok állnak szemben az ellenfél hasonló erejével.

A tapasztalat azt mutatja, hogy a végjátékok fele bástyavégjáték, melynek közvetlen célja rendszerint valamelyik gyalog bevezetése, és csak ritkán a király gyors mattolása. A bástyavégjáték tehát gyakran gyalogvégjátékká alakul át.

A bástyavégjátékban néha nagyobb anyagi előnnyel szemben is jó eséllyel küzdhetünk a kiegyenlítésért. Ilyenkor nagy jelentősége van az aktív bástyának és királynak. Néhány dolgot feltétlenül szem előtt kell tartanunk:

1. Igyekezzünk elvágni ellenfelünk királyát a gyalogjától, mert ezzel a gyalog bevezetése lényegesen megnehezül, gyakran lehetetlenné is válik.

2. A védekező király elválasztása előrehaladó gyalogunktól megkönnyíti a gyalog bevezetését.

Ezeket a meggondolásokat viszonylag könnyen beprogramozhatjuk. Először is meg kell vizsgálni, melyik oszlopban van a király és melyikben a gyalog. Ha nem a szomszédos vagy azonos oszlopban, akkor az e két oszlop közötti mezőket nagyobb súllyal vegyük figyelembe bástyalépés esetén, mint addig.

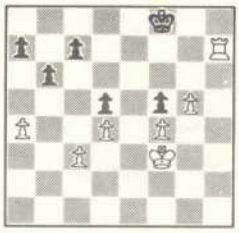
3. Az előrehaladó gyalogot legjobban a mögötte elhelyezett bástya támogatja; ilyenkor a gyalog előrenyomulásával nő a bástya mozgékonysága is.

4. A védekező fél bástyája is az előrehaladó gyalog mögött fejtheti ki legelőnyösebben tevékenységét.

E szempontokat is egyszerű a programban figyelembe venni: a bástya számára fel kell pontozni a szabad gyalog(ok) mögötti mezőket. Ugyancsak érdemes a programban érvényesíteni, hogy a világos bástya a 7. soron, és ennek megfelelően a sötét bástya a 2. soron sok taktikai veszélyt jelent!

Az aktív bástya és király gyaloghátránnyal sokszor többet ér, mint a gyalogelőny passzív bástya és király mellett. Az aktivitás fokozására gyalogáldozatokat is érdemes hozni, amit klasszikusan példáz Capablanca 1924-ben Tartakover ellen nyert játszmája. A következő hadállásból indulunk: világos két gyalogját is ütni engedi, hogy bástyájának aktív helyzetét királyának támogatásával kihasználhassa. Következett:

35. Kf3—g3! Bc6xc3
36. Kg3—h4 Bc3—f3
37. g5—g6! Bf3xf4+
38. Kh4—g5 Bf4—e4
39. Kg5—f6!



Láthatjuk, hogy egyik fél sem üti a másik lógó gyalogját. Ehelyett világos támad, sötét védekezik.

39. — Kf8—g8
40. Bh7—g7+ Kg8—h8
41. Bg7xc7 Be4—e8
42. Kf6xf5 Be8—e4
43. Kf5—f6 Be4—f4+
44. Kf6—e5 Bf4—g4
45. g6—g7+ Kh8—g8

A gyalog ütése utáni csere és gyalogvégjáték természetesen veszít sötét számára Kd5:—Kc6 stb. miatt.

46. Bc7xa7 Bg4—g1
47. Ke5xd5 Bg1—c1
48. Kd5—d6 Bc1—c2
49. d4—d5 Bc2—c1
50. Ba7—c7 Bc1—a1
51. Kd6—c6 Ba1xa4
52. d5—d6 Sötét feladta.

Tanulság: az ellenfél bástyasakkjai ellen csak akkor védekezzünk bástyánkkal, ha a csere után előálló gyalogvégjátékot kedvezőnek ítéljük a magunk számára. Ez olyan alapelv, amelyet programban még nem tudtak érvényesíteni. A programozók ugyanis csak néhány esetben képesek végérvényesen átgondolni, hogy a csere előnyös-e vagy sem. Mégis akad egy-két támpont ennek a problémának a megközelítéséhez:

— ne cseréljünk, ha anyagi hátrányunk van;

— ha szabad gyalogunk van, a csere előtt állapítsuk meg, hogy elegendő távolságban van-e az ellenfél királyától a bevitelhez;

— egyes programokban e nehézséget transzpozíciós táblák használatával próbálják áthidalni. Ezekről a megoldásokról még szó lesz.

A programot egyszerűen „meg is taníthatjuk" arra, hogy egy bástyával és királlyal az ellenséges királyt bemattolja. Nem véletlen, hogy az első gépi program éppen erre volt képes. Ha a standard centrumsúlyozású táblát használjuk, a következő képlet elég hatásos:

[16—abs(Δoszlop)
+abs(Δsor)]+
+2*(08-ellenséges
kir. centrumértéke)
+oppozíció

A program így megtalálja a mattot. A képlet első része biztosítja, hogy a királyok közeledjenek egymáshoz, a második tagja által pedig az erősebb fél bástyája leszorítja a gyengébb fél királyát. Az oppozíciós tagnak köszönhető a királyok kényszerű szembenállása, ami segít az ellenfél királyának leszorításában.

Amilyen könnyű egy sakkprogramnak egy bástyával a matt elérése, olyan nehéz két bástyával és királlyal egy bástya plusz király ellen harcolni. Bár tudjuk, hogy ez az állás egy-két kivételtől eltekintve nyer, a program sokszor nem képes kihasználni anyagi előnyét, előnyös pozícióját, és a játszma döntetlennel végződik. Elméletileg olyan stratégiának kellene érvényesülnie, mely szerint az erősebb fél megpróbálja lecserélni egyik bástyáját az ellenfél bástyájáért. Így visszavezethetnénk a problémát az előző feladatra, amelyet megoldottunk. De hogyan értessük meg ezt a számítógéppel?

Nem nehéz olyan programot írni, amelyben az erősebb fél kísérletet tesz az egyik bástyájának lecserélésére úgy, hogy ezt mindig ütésbe teszi olyan mezőn, amelyet királyával vagy a másik bástyájával ellenőriz. Sokkal nehezebb megoldani, hogy ne csak ütésre játsszon, mert ha az ellenfél ismeri a problémát, úgysem üti le. Ki kell kényszeríteni a bástyák lecserélését. Ezt úgy érhetjük el, hogy vagy mattra állítjuk az ellenséges királyt, vagy lekötjük a bástyáját. Nagyon nehéz azonban az erre vezető matematikai képletet megtalálni. Kísérletezéseim eddigi eredményének ismertetése sem jelentene az olvasónak hathatós segítséget. De ne adjuk fel!

KOVÁCS P. ATTILA

µINFORM

Mikroinform
A tartalomleírások az alábbi folyóiratokban megjelent programlistákról készültek:

A folyóirat neve	Kódja
64'er Magazin	64er
Chip Magazin	chip
Commodore Horizons	coho
Commodore Microcomputers	comi
Compute!	cute
Computer Persönlich	pers
Happy Computer	happ
hc - Mein Home-Computer	hc
mc - Zeitschrift	mc
Run /USA/	run
Sinclair User	sinc
Your Sinclair	ysin

A tartalomleíró szólláncokat permutáltuk, a szólánc-variánsokat pedig alfabetikusan rendeztük.

A tartalomleírás egy szólláncból áll, majd a listában ezt követi a forrás megjelölése a folyóirat azonosítójával, a megjelenés dátumával és a cikk előkeresésével a kezdő oldalszám és a terjedelem megadásával. A mellékelt lista értelmezéséhez még az alábbiakat kell tudni. A tartalomleírás szólláncában elsőként a téma átfogó megnevezése, utána a számítógéptípus(ok), ezt követően a szűkebben jelölt tartalom meghatározása szerepel, majd esetlegesen néhány, a közleményt minősítő adat (például : cikksorozat).

A forráshely karaktersorozatát nyíl vezeti be, melyet a / jelig a folyóiratok azonosítója, a két / jel között az évszám, folyóiratszám és kötőjellel a kezdő oldalszám követi, a végén pedig a közlemény teljes oldalterjedelme áll.

A folyóiratok a SZÁMALK szakkönyvtárában (Budapest XI., Szakasits Á. út 68. Nyitva: 8-tól 'fél 5-ig. Tel.: 853-111/251) is föllelhetők. A kiválasztott anyagról másolat rendelhető az alábbi formában:

SZÁMALK Szakkönyvtára
Budapest, 112. Pf.: 146. 1502

Megrendelem a Mikroszámítógép Magazin 1987/ sz. alapján a következő folyóirat-oldal-másolatokat:

Kód: _____ Példányszám: _____
Kód: _____ Példányszám: _____
Kód: _____ Példányszám: _____

A megrendeléshez csatolom az oldalankénti 8,- Ft-os szolgáltatási díj befizetését igazoló csekkszelvényt.
Dátum, név, pontos cím.

PROGRAMLISTA
apple ii▯atari 400/800 xl/xe▯commodore 64▯oktatas▯jatekos ora-ido leolvasas ->cute/86.05-36/6
PROGRAMLISTA
apple ii▯kiegeszites a (85.06)54-hez ->hc/86.05-76/1
PROGRAMLISTA
applesoft▯goto es gosub modositas el agaztatashoz ->cute/86.05-75/2
PROGRAMLISTA
applesoft▯modositott escape katalogusmegjelenitéshez ->cute/86.05-93/2
PROGRAMLISTA
atari 400/800 xl/xe▯basic programozas▯hibamentesites wedge-technikaval ->cute/86.05-81/5
PROGRAMLISTA
atari 600/800xl/130xe▯monitorprogramkent futtathato zenei alafestes ->hc/86.05-81/1
PROGRAMLISTA
atari 800 xl▯cimkor-generalas a 24 sorig ->happ/86.05-83/1
PROGRAMLISTA
atari st▯basic programozas▯cikksorozat▯(st mouse pointer) ->cute/86.05-65/4
PROGRAMLISTA

atari st▯muveletgyorsitas▯tos lekerdezes kiiktatasa ->happ/86.05-82/2
PROGRAMLISTA
atari xl/st▯listabegepelesi segedlet ->hc/86.05-80/1
PROGRAMLISTA
barkacsolas▯commodore 64▯ellenallaserterk-szingyuru tablazat elektrotechnikahoz ->happ/86.05-60/1
PROGRAMLISTA
cikksorozat▯commodore 128▯grafika(ultra-hires)▯specialis programnyelv ->run/86.05-34/8
PROGRAMLISTA
cikksorozat▯commodore 64▯grafikus programozas▯matematika▯matrixszorzas ->64er/86.05-145/7
PROGRAMLISTA
cikksorozat▯commodore 64▯modulbetoltes es kapcsolas a futtatott programhoz ->hc/86.05-70/1
PROGRAMLISTA
commodore 128▯(auto boot 128) ->happ/86.05-57/1
PROGRAMLISTA
commodore 128▯grafika▯(128 drawing tablet) ->comi/86.05-140/8
PROGRAMLISTA
commodore 128▯grafika▯matematika ->happ/86.05-54/3
PROGRAMLISTA
commodore 128▯jatekprogram▯(rotation tag) ->comi/86.05-134/3
PROGRAMLISTA
commodore 16;plus/4▯felhasznaloi Karakterek ->cute/86.05-14/1
PROGRAMLISTA
commodore 64▯(64 autobooter) ->cute/86.05-90/3
PROGRAMLISTA
commodore 64▯(disc-wizard)▯lemezegysleg(1541)-kezeles ->64er/86.05-54/8
PROGRAMLISTA
commodore 64▯(disk reader)▯fileszerkesztes ->run/86.05-58/3
PROGRAMLISTA
commodore 64▯(label maker)▯cimke szerkesztes/sokszorositas ->run/86.05-56/3
PROGRAMLISTA
commodore 64▯(mc-screen)▯sorokra bonthato keret/hatter szinbeallitas ->hc/86.05-76/3
PROGRAMLISTA
commodore 64▯(profi-text)▯irogep-helyettesites ->chip/86.05-141/4
PROGRAMLISTA
commodore 64▯comemodore microcomputers▯listabegepelesi segedlet ->comi/86.05-170/2
PROGRAMLISTA
commodore 64▯data-sor bevitel csak felsosori billentyukkel ->run/86.05-69/1
PROGRAMLISTA
commodore 64▯grafika▯(super-hardcopy)▯ml-rutinok kulonbozo nyomtatokhoz ->64er/86.05-70/8
PROGRAMLISTA
commodore 64▯grafika▯matematika▯fuggvenyabrazolas▯vonalrajzolas epson nyomtatoval ->64er/86.05-79/3
PROGRAMLISTA
commodore 64▯grafika▯matematikai osszefuggesek alkalmazasa▯peldak ->hc/86.05-71/2
PROGRAMLISTA
commodore 64▯grafika▯meret/pontsurusseg valtoztatas▯(super-print) ->64er/86.05-63/6
PROGRAMLISTA
commodore 64▯grafika▯nagyitott Kepernyojelek nyomtatasa▯(greatprint) ->64er/86.05-69/1
PROGRAMLISTA
commodore 64▯hypra-basic▯sprite-Kezelo modulok ->64er/86.05-103/4
PROGRAMLISTA
commodore 64▯interrupt programozas▯(basic-int) ->64er/86.05-103/1
PROGRAMLISTA
commodore 64▯jatekprogram▯(speed boat race) ->comi/86.05-62/3
PROGRAMLISTA
commodore 64▯jatekprogram▯(steel slab) ->64er/86.05-86/2
PROGRAMLISTA
commodore 64▯jatekprogram▯Kosarlabdajatek Ket botkormannyal ->run/86.05-40/4
PROGRAMLISTA
commodore 64▯programreszletek Kulon tarolasa a lemezen ->hc/86.05-70/2
PROGRAMLISTA
commodore 64▯sajat KarakterKeszlet szerKesztese lemezre ->64er/86.05-81/5
PROGRAMLISTA
commodore 64▯vesszo es Kettospont megengedese inputhoz ->comi/86.05-151/5
PROGRAMLISTA
commodore 64:128▯havi Kalendarium elloallitasa 1900 es 2000 Kozott ->happ/86.05-62/4
PROGRAMLISTA
commodore 64:128▯jatekprogram▯(quadrophenia) ->happ/86.05-52/3
PROGRAMLISTA
commodore 64:128▯numerikus output elloallitas string funKcioKKal ->run/86.05-70/4
PROGRAMLISTA
compute!▯listabegepelesi segedlet ->cute/86.05-100/3
PROGRAMLISTA
happy computer▯listabegepelesi segedlet(mse) ->happ/86.05-59/2
PROGRAMLISTA
helyesbites▯(86.01)43;44;144(86.03)79;84(86.04)48;84 ->64er/86.05-87/1
PROGRAMLISTA
lemezegyseg(1541;1571)▯(disK utility)▯uzemeltetesi tanacsoK ->comi/86.05-156/7
PROGRAMLISTA
matematika▯pascal programozas▯linearis egyenletek ->hc/86.05-64/3
PROGRAMLISTA
rejtjelezes▯sinclair spectrum▯(geheimschrift) ->hc/86.05-49/2
PROGRAMLISTA
run▯1985-ben Kozolt beviteli segedlet-valtozatok ->run/86.05-102/4
PROGRAMLISTA
sinclair spectrum▯cikksorozat▯jatekprogram Keszites ->zx/86.05-35/3
PROGRAMLISTA
sinclair spectrum▯grafikus ml-rutin teruletKitolteshez ->hc/86.05-40/2
PROGRAMLISTA
sinclair spectrum▯jatekprogram▯(banana-trouble) ->hc/86.05-45/4
PROGRAMLISTA
sinclair spectrum▯jatekprogram▯(lord of darkness) ->zx/86.05-26/7
PROGRAMLISTA
sinclair spectrum▯jatekprogram▯(mega-drive) ->zx/86.05-84/3
PROGRAMLISTA
sinclair spectrum▯masodiK hang generalasa ->hc/86.05-42/4
PROGRAMLISTA
sinclair spectrum▯ml-rutin tesztelés ->zx/86.05-78/3
PROGRAMLISTA
sinclair spectrum▯orageneralas/aKtivalas ->happ/86.05-84/1
PROGRAMLISTA
speedcalc▯apple ii▯commodore 64:128▯extra-szeles tablazatoK Keszitese ->cute/86.05-10/2
PROGRAMLISTA
speedscript▯atari 400/8000 xl/xe▯felhasznaloi KarakintereK generalasa ->cute/86.05-80/3
PROGRAMLISTA
szovegfeldolgozas▯commodore 128▯(archetype) ->comi/86.05-146/5
PROGRAMLISTA
veletlenszam▯commodore 128▯muveszi grafika▯absztrakt abraK Keszitese ->comi/86.05-57/1

Avarosi László, Budapest,

Hámán Kató út 4. 1096

Szinte hihetetlen, hogy az ország legolvasottabb számítógép-magazinjának a szerkesztősége semmiféle gépparkkal nem rendelkezik. Ez derül ki ugyanis az olvasói rovat elején található intelemből, mely szerint kérik a leendő publikálókat, hogy programjaikat ne adathordozón, hanem nyomtatott lista alakjában küldjék be. Feltehetően — eszközök híján — a szerkesztőségnek nem áll módjában, hogy az adathordozókat kilistázza. Talán ez is oka annak, hogy néha elcserélt vagy hiányos listamellékletek látnak napvilágot.

Szerintem a probléma megoldható olyan segítőkész hobbi-számítógépesek közreműködésével, akik a rendelkezésükre álló hardverrel szívesen támogatnák az Önök munkáját. (. . .)

Noha megvan a jelentősége a kész programok közlésének, mégis a „gépfüggetlen" írásokat tartom igazán közérdekűeknek. Sajnos ezek némelyike olykor semmitmondó. Igen hasznos volna egy általános érvényű, algoritmizálási alapgyakorlatokból álló cikksorozatot készíteni a Programozástechnika című rovatban, amely a problémameglátás és -megoldás képességét fejlesztené egy-egy érdekes példa segítségével. Változatosabbá tenné a magazint egy szórakoztató (fél)oldal, ahol az ügyesen fogalmazók elmesélhetnék közérdeklődésre érdemes, humoros vagy tanulságos számítógépes történeteiket, viszontagságaikat.

Végül megemlítem, hogy az utolsó két szám (87/1—2.) címlapi grafikája igen jól sikerült, a nagy címfelirat is előnyösebb.

Valóban csak egy-egy számítógép áll rendelkezésünkre. Nyomtatónk pedig csak egy van (és az is csak egyetlen géppel képes együttdolgozni).

A „segítőkész hobbi számítógépesek" helyzetét ismerem, és tudom, hogy náluk is hiánycikk a nyomtató. Amennyiben Ön segítene, szívesen fogadjuk. A „közlés előtti ellenőrzést", ha nem is rendszeresen, de végzik. A „gépfüggetlen" írásokkal kapcsolatos javaslatát már több sorozatunkban is megvalósulni láthatja. Ilyen például a „Játékprogramozás technikája", „Integrált szoftverek", „Teknősbéka-grafika BASIC-ben".

Humoros írásokból sajnos nem kapunk eleget. Ami elfogadható színvonalú, közöljük.

Szabó István, Budapest,

Néphadsereg tér 10/B 1055

Rendszeres olvasója vagyok a Magazinnak, és mindeddig abban a hitben éltem, hogy Önök is olvassák a lap nyomdából kikerült példányait is, és levonják a levonható tanulságokat. A tapasztalat azonban nem ezt mutatja, és nálam betelt az a pohár, melynél hangos kifakadással szoktam jelezni nemtetszésemet.

A lapban közölt programlisták olvashatósága a kiváló és sajnos a teljesen olvashatatlan között mozog. Én tudom, hogy rengeteg forrásból kapnak listákat, ilyen-olyan nyomtatókkal, festékszalaggal írva, ezért erre ne is hivatkozzanak, ezt a gondjukat megértem. Azt azonban már nem, hogy a tisztelt tördelőszerkesztő miért nehezíti az életet azzal, hogy kék alapra nyomtattatja rá ezeket a listákat, ezáltal a gyengébb minőségű listák olvashatóságát nullává téve, különösen azokét, amelyek még nyomdatechnikai úton is vannak kicsinyítve. (. . .)

Végül gratulálok a magas szintű tartalmi munkához. S lenne még egy javaslatom: az esetleges nyomdahibák, szemantikai hibák korrigálására nyissanak külön rovatot (esetleg egy kis ördög emblémával), amely vagy van, vagy nincs az aktuális lapszámban. Ez megkönnyítené az olvasók dolgát. (. . .)

Kritikájával egyetértünk. Természetesen mi is szeretnénk elérni, hogy csak jól olvasható listák jelenjenek meg lapunkban, de ez esetben igen sok, érdeklődésre számot tartó programról kellene lemondanunk a rossz minőségük miatt. Tervezőszerkesztőnk szerint egyébként a kék alapszín nem rontja, hanem éppen ellenkezőleg, javítja a halvány listák olvashatóságát.

ifj. Fábián János, Heves,

Klapka út 36. 3360

A közelmúltban játékkazettákat kaptam, melyeket nem tudok betölteni. A kazetta papírján a következő áll: a sikertelen betöltés esetén a HEADJUST fejbeállító programmal végezze el a magnófej finombeállítását. Sajnos ezt nem tudom elvégezni, mert nem ismerem a használatát. Kérem segítsen ebben!

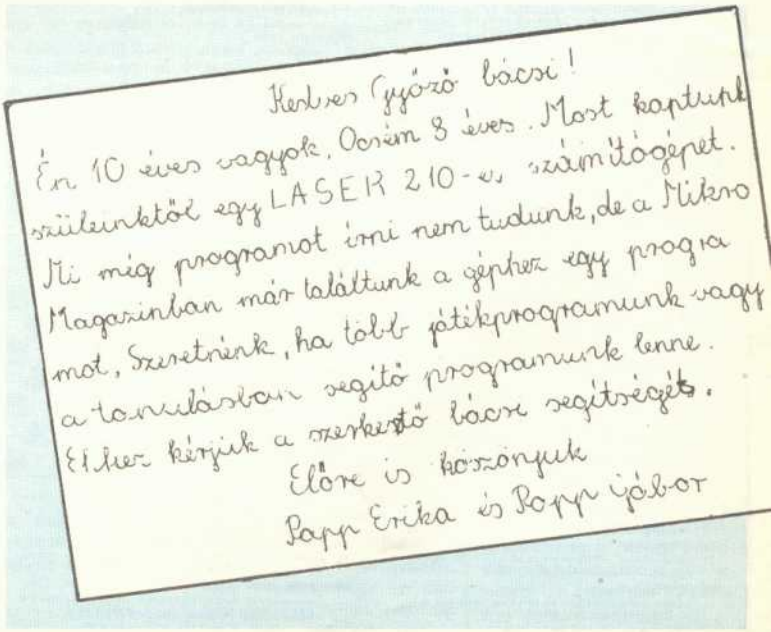
Gyakran kapunk olyan olvasói kérést, cikket, amelyeken nem szerepel, milyen típusú számítógéppel kapcsolatos. *Ez olyan alapvetően fontos információ, ami nélkül általában nem tudunk válaszolni.*

Sok olvasónktól kapunk árakról, beszerzési helyekről kérdéseket. Még olyat is, hogy milyen mértékű lesz a Spectrum leárazása és lehet-e OTP hitelre vásárolni? Ezekre általában nem tudunk válaszolni. Az általunk ismert beszerzési helyeket — esetleg árakat — kérés nélkül is igyekszünk közölni.

Lukács Attila, Jesenského 68.

943 01 Sturovo

(. . .) A C64-esnél jöttem rá, hogy mi is az a számítógép (bár lehet, hogy már kissé elfogult vagyok vele szemben). Ami a C64 programellátását illeti nálunk, az szinte ka-



tasztrofális, nem lehet egy normális programot szerezni. A gépi kódú programozást segitő programokat én irtam (assembler, monitor), s grafikát csak a SIMON'S BASIC-kel tudok csinálni.

Érdekelne, hogy lehetne a C64-hez házilag összehozni egy fényceruzát? A VIC-nek van LP bemenete, s két regiszter tájékoztat az X, Y koordinátákról. A SIMON'S BASIC-nek is van PENX, PENY függvénye. Továbbá az érdekelne, hogyan lehet az LP-t szoftver úton használni (valami rövid demo program, ha lehet gépi kódban).

S még egy kérdés: nálunk is lehet kapni teljes ATARI konfigurációt (gép, floppy, nyomtató, tablet, egér), ezek közül lehetne-e valamit csatlakoztatni a C64-hez (mondjuk az egeret)?

Úgy tűnik, hogy talán nem is csak „kissé" elfogult. Szeretném emlékeztetni a 85/6. szám „Mit ér a C16?" című cikkre, a 86/10-esben a Plus/4 termékismertetőjére, természetesen nem azért, hogy elvegyem a kedvét kedvenc gépétől . . . Nagyon elterjedt az a felfogás, hogy: csak az én gépem lehet jó, a többi nem. Az ilyen beállítódás megakadályozhat valakit abban, hogy a gépeket reálisan értékelve az optimálisat válassza ki, illetve a legcélszerűbben használja az elérhető gépet.

Konkrét kérdéseire: itt mindkét eszközt elfogadható áron láttam a μ '87 kiállításon (a fényceruzát 2500,— az egeret 4400,— Ft árért). Úgy vélem, hogy — amennyiben ezeknek nemcsak az ára, de a minősége is elfogadható — nem érdemes kísérletezni.

Az Atari perifériákat átalakitók nélkül nem lehet használni.

A szerkesztőség fontosnak tartja értesíteni a cikket küldő olvasókat, hogy a továbbiakban a következőknek meg nem felelő cikkjavaslatokkal nem foglalkozunk:

1. A cikk leíró része írógéppel, egy oldalon 30 sor és 60 leütéssel van írva. Az esetleges mellékletekre (lista, ábra) a szövegen belül egyértelmű hivatkozás van.
2. Névvel, címmel és személyi számmal ellátott írásokat kérünk.
3. A mellékelt listát (listákat) kinyomtatva, soronként legfeljebb 40 normálméretű karakterből összeállítva kérjük. Csak jól olvasható listákkal tudunk foglalkozni. A listát kérjük úgy összehajtani, ha ez elkerülhetetlen, hogy a hajtás ne törje meg a sorokat.
4. Az ábrák félre nem érthetőek, feliratozásuk világos, arányaik érzékletesek és megfelelnek a szemléltetendő dolog közérthető bemutatásának.

Mindezt nem a leendő szerzők felesleges zaklatásáért kérjük, hanem azért, hogy lerövidítsük a cikk közölhető formába öntésének folyamatát.

Homonnay Péter: Angol—magyar számítástechnikai szótár (Budapest, 1986. Novotrade Rt., 136 oldal. Ára: 74,— Ft.)

A számítástechnika legbővebb forrásnyelve még mindig az angol: a számítógéphez tartozó leírások is jórészt csak angolul jelennek meg.

A szótár elsősorban a TPA—11 számítógépcsalád felhasználói programozóinak kíván segítséget nyújtani, de az ESZR és a személyi számítógépekkel dolgozók is haszonnal forgathatják. A kötet a számítástechnikában leggyakrabban előforduló szakkifejezéseket tartalmazza néhány közhasznú szó kiegészítésével. Az angol kifejezések magyar megfelelőit az eredeti szövegkörnyezet alapján válogatta ki a szerző.

C16 PLUS/4 programozói útmutató (Budapest, 1987. Novotrade Rt., 84 oldal. Ára: 129,— Ft.)

A kézikönyv mindazokat az információkat tartalmazza, amelyekre a programozás közben szükség lehet. A programok elkészítéséhez a kezdő programozó éppúgy megtalálja benne a szükségeseket, mint a gépi kódú programozásban (a 6502 gépi nyelvben) jártas profi.

A kötet sikeres használatának csak egy feltétele van, hogy az olvasónak legyenek számítástechnikai alapismeretei. Ezek elsajátítására a szerzők az adott számítógép felhasználói ismertetőjét javasolják.

A kézikönyv először a BASIC 3.5 programozási nyelvvel foglalkozik. Ez az ismeretár lexikonszerűen sorolja fel a BASIC utasításokat, parancsokat és függvényeket, és tartalmazza valamennyi kulcsszó megengedett rövidítését is.

A Commodore gépek grafikai lehetőségeire vonatkozó információkat is megtalálja az olvasó.

A második rész tulajdonképpen a gépi kódú programozás részletes leirása.

A függelékben különböző hasznos, technikai tudnivalók találhatók (képernyőkódok, ASCII és CHR$ kódok, képernyőtár- és szintár-térképek stb.).

Grohmann, B. — Eichler, L.: A 68000-es mikroprocesszor. Technika és programozás (Budapest, 1986. Data Becker — Novotrade, 363 oldal. Ára: 349,— Ft.)

A kötetben a Motorola mikroprocesszorcsaládról szóló szinte valamennyi ismeret megtalálható. Elsősorban abban kívánja az olvasót segíteni, hogy miként használja ezt a processzort és hogy érthetővé tegye az alkalmazási területeit. Bemutatja magát a hardvert is.

A könyv főbb fejezetei: A 68000-es mikroprocesszor felépítése — Jelek és buszok a 68000-esben — A család további processzorai — A 68000-es perifériái — Az utasítások és címzésmódok — Az utasításkészlet — A 68000-es processzor az operációs rendszerekben — Programozási példák.

COMPUTER—S

Közel egy éve kezdte meg a Centrum Áruház országos hálózatában árusítani a Videoton TV Computerét, decemberben pedig a Skála lépett egy nagyot előre: az ország 22 Skála-áruházában számítógépes sarkot alakított ki. A háziszámítógépektől a professzionális berendezésekig s az ezekhez szükséges kellékeket, alkatrészeket egy helyen kezdték árusítani. Karácsonyra külön felkészültek, s négyezer háziszámítógépet értékesítettek kilenc-tíz ezer forint közötti áron.

A Computer—S(arok) eladóit rendszeresen továbbképzik, hiszen az ő szakértelmük nem közömbös az értékesítés eredményességében. A Computer—S célkitűzései között nemcsak az árusítás, hanem a számítógépes kultúra terjesztése is szerepel.

Ítél a gép a felvételin

Az idei szeptembertől két tannyelvű képzést indít hét gimnázium. A jelentkezők — csaknem háromezer tanuló — januárban tettek felvételi vizsgát. Az összesen 12 osztályba pályázók írásbeliztek: a vizsga során logikai, matematikai gondolkodásukat, kreativitásukat bizonyíthatták. Az elbírálók az értékelési módszerek alkalmas megválasztásával felmérték azt is, hogy az adott szövegből, szemelvényből a jelöltek megértik-e az információt, hiszen ez a tanulás alapja. A felvételi vizsga során kitöltött feladatlapok rovatait számítógépbe vitték. A számítógépes feldolgozás minősítette a megoldásokat, és az elért eredmények alapján a gép rangsorolta a pályázókat. A felvételi bizottság ennek figyelembevételével könnyebben végezhette el munkáját, s az egész folyamat demokratikusabbá vált.

A vakokért

A hazai Homelab háziszámítógépekből még tavaly elkészült a vakoknak szánt Brailab gép, mely a begépelt adatokat, programokat szintetizált hangon magyarul „beszélve" is visszajelzi. A gép tökéletesítéséhez az idén a Soros Alapítvány is hozzájárult, lehetővé téve tőkés fejlesztőrendszer beszerzését.

A Brailab azonban már a jelenlegi állapotában is igen sok vak számára könnyítheti meg a mindennapokat. Ezt a tavaly elkészült gépek alkalmazási tapasztalatai is bizonyítják. (A Tudományszervezési és Informatikai Intézet anyagi támogatása révén különböző, vakokkal foglalkozó intézményekhez jutottak el az első darabok.)

A fejlesztők az idén ötven darab készítésére vállalkoznak. A fedezet biztosítására — a nem látó kollégák iránti tiszteletből is — a zenészek fellépéseket vállaltak, előadásokat szerveztek. A rendezvények februárban kezdődtek és áprilisban, a Petőfi-csarnokbeli koncerttel értek véget. A teljes bevételt a Brailab gépek gyártására fordítják.

Leáldozóban a szakácskönyvek karrierje?

A svájci Provaltec AG új terméke, a Merlin nevű konyhai számítógép az első kérdésre — Mit főzzek ma? — 500 címszó alapján válaszol. Az étlap összeállításakor figyelembe veszi az évszakot, a napszakot, az étkező személyek számát és az elkészítés nehézségi fokát. A gép különálló fogásokat vagy teljes étlapokat javasol, és mennyiségi variánsokat tud összeállítani: két személytől 12 személyig képes az adagok meghatározására. Mindehhez igény szerint mellékeli a hozzávalók listáját, a recepteket, a tápértékadatokat, a tálalás módját. Akár képernyőn megjelenítve, akár a hozzá csatlakoztatott hőnyomtatón kinyomtatva — bevásárlócédulaként hasznosíthatóan — kaphatjuk meg a gép ajánlatait.

A Merlin kezelése igen egyszerű. Felszerelhető rögzítve is, de hordozható változata is van, mert a tápellátása az elektromos hálózattól független. Ára a mosogatógépé és a mikrohullámú sütőé közé esik.

Villamosgépek tervezése korszerűen

A csehszlovákiai Erősáramú Villamos Berendezések Tröszt jelentőségét egyetlen adat is tükrözi: 60 ezer dolgozója van. Három kutatóintézetének egyike a brnói Forgóvillamosgépek Kutatási és Fejlesztési Intézete, itt tervezik Csehszlovákia összes üzeme részére a forgóvillamosgépek prototípusát.

A 620 munkatársat foglalkoztató brnói kutatóhely — amelynek Kassán is van egy 320 fős fiókintézete — kapcsolatban áll a KGST-országok hasonló kutatóintézeteivel is, és nagy súlyt helyez a konstruktőrök és a technikusok munkájának hatékonyabbá tételére. Ma már a mérnöki tervezési munkákat is számítógépesítették. Az ún. grafikai munkahelyen 30-40 perc alatt már meg lehet tervezni egy-egy új forgógépet, motort, s utána automatikusan elkészülnek a részletes gyártási tervrajzok is. Ezt követően a számítógéppel gyorsan lefutnak az elektromágneses, a hőszolgálati, a mágneses, a szellőztetési és a mechanikus számítások, ellenőrizhetők és korrigálhatók az előző tervezési fázisok. Mindez a klasszikus módszerrel 300, sőt több órát is igénybe venne.

Az eddigi sikerekre alapozott elképzelések szerint 1990-re már alig lesz rajzasztal az intézetben, az asztalok és a rajzolók helyét a számítógépek foglalják el.

Vezetékek nélkül

A GridComm Inc. nevű vállalkozás bejelentése, miszerint megvalósította a sokak által hőn áhított célt, a hálózati dugaljakra csatlakozó számítógépes hálózatot, igazi szenzáció. Állításuk szerint a telepítés költségei a felére csökkennek az adatátviteli kábelezés elmaradásával. A kérdés csupán az, hogy lehetséges-e az energetikai hálózaton valóban megbízható, hibátlan adatátvitel? A megoldás kulcsa egy újszerű modulációs eljárás, amelyet a cég a saját nevével fémjelez: GCM vagyis GridComm Moduláció. A szabadalmi eljárás folyamatban van.

A GCM eljárás 23 kbit/s sebességű, teljes duplex átvitelt biztosít az átviteli folyamatban az egybites hibák maradéktalan javításával. A modulációs eljárás leküzdi a vivő zavaraiból, illetve kimaradásából származó problémákat.

Az erősáramú vezetéken történő adatátvitel előnye nyilvánvaló: az adatátviteli kábelezés költsége tetemes, a szerelési munkaigénnyel együtt bizonyos esetekben a kiszolgálandó mikrogépek, illetve perifériák értékének akár 75%-át is elérheti. A GCM-állomások viszont a kiszolgálandó gép, illetve periféria értékénél csak 20%-kal drágábbak.

Hasonló jellegű rendszert eddig csak az angol Apex Communications Ltd. kínált, az erősáramú vezetéken félduplex adatátvitellel. Ez a Nectar Ring, melynek azonban a 4800 bit/s átviteli sebessége meg sem közelíti a GridNet-et.

A GridNet mintegy 3000 m^2 alapterület ellátására képes közönséges egyfázisú hálózaton. Egy helyi hálózat 60 címezhető csomópont kiszolgálására alkalmas, de egyszerre mindig csak 8 résztvevőt enged szóhoz jutni. Ezért egyelőre a legfeljebb 32 résztvevős hálózat látszik optimálisnak. A fejlesztési elképzelések között elöl szerepel az átviteli frekvenciasáv kiszélesítése, amivel az egyidejűleg kiszolgálható állomások száma a kétszeresére emelkedik. A hálózat minden résztvevőjéhez két egységből álló illesztőállomás csatlakozik. Ezek megnevezése és ára:

GC — 1400 személyi számítógépekhez 549 USD
GC — 1100 perifériákhoz 449 USD
GC — Zero telefonvonalakhoz 799 USD

Az utóbbiból látható, hogy a nagy távolságú adatátvitel lehetősége is fennáll.

A hálózat az IBM PC-vel kompatibilis gépeken kívül az Apple, a DEC, a HP, a Tandy és a Wang mikrogépeit is fogadni tudja. Illeszkedik ezenkívül majdnem minden soros és párhuzamos illesztőjű nyomtatóhoz és rajzológéphez.

VIDEOTON VÁLASZTÉK
OPTIMÁLIS VÁLASZTÁS

A VIDEOTON új professzionális személyi számítógépei, a VT 110 és VT 160, a világszabványba illeszkedő IBM PC/XT és AT-vel kompatibilis számítógépek.

A gépek csatlakoztathatók az eddig ismert VIDEOTON rendszerekhez, elemei lehetnek számítógép-hálózatnak, további lépcsőfokot jelentenek a fejlődéshez.

Nemcsak a közvetlen munkahelyi környezetben végzendő munkákat — szövegszerkesztés, adatrögzítés, kalkuláció — könnyítik meg, hanem lokális postai hálózaton keresztül bekapcsolhatók nagy rendszerekbe is.

A VT 110 és 160-as számítógépekkel többmunkahelyes rendszerek kialakítása lehetséges:

— a VT 160-as gépekhez display-munkahelyek csatlakoztathatók,

— a VT 110 és VT 160-as gépekből lokális hálózat alakítható ki.

Kompatibilitásából következik, hogy az IBM vagy IBM kompatibilis személyi számítógépekre kidolgozott programok futtathatók az új gépeken.A VT 110, IBM PC/XT kompatibilis 640 kbájt memóriával rendelkező alapgép fekete-fehér monitor, magyar ékezetes tasztatúra, 360 kbájt kapacitású floppy.

Választható opciók:

— 10 és 20 Megabájtos Winchester diszk

— nagy felbontású színes monitor

— 80 vagy 132 oszlopos magyar ékezetes mátrixnyomtató

— NLQ minőségü mátrixnyomtató

— további floppy bövítés

— streamer

— MS-DOS 3.2 operációs rendszer

VT 160, IBM PC/AT kompatibilis, 640 kbájt memóriával rendelkező alapgép — fekete-fehér monitor, magyar ékezetes tasztatúra, 1,2 Mbájt kapacitású floppy.

Választható opciók:

— 20 és 40 Megabájtos Winchester diszk

— memóriabövítés 2,5 Mbájtig

— 4 és 8 vonalas soros interfész

— nagy felbontású színes monitor

— 80 vagy 132 oszlopos magyar ékezetes mátrixnyomtató

— NLQ minőségü mátrixnyomtató

— streamer

— MS-DOS 3.2 operációs rendszer

Teljes körű szolgáltatásunk, az országos VIDEOTON vevőszolgálat a VT 110 és VT 160-as professzionális személyi számítógépek vásárlói számára is rendelkezésre áll.

VIDEOTON
ELEKTRONIKAI VÁLLALAT

MINDEN,
AMIT TUDNI KELL
AZ IBM PC
XT/AT-RŐL!
SZÁMSZÖV
SZÁMÍTÁSTECHNIKAI KISSZÖVETKEZET
Cím: Bp. XI., Hunyadi János u. 162. 1116. Levélcím: Bp., Pf.: 16. 1430.
Telefon: 665-656, 665-322, 131-072